sat-nms Software User Manual

User Manual

Version 3.80.6-a -- 2025-12-03

© Copyright
SatService Gesellschaft für Kommunikationssysteme mbH
Hardstrasse 9
D-78256 Steisslingen

satnms-support@satservicegmbh.de

www.satnms.com
www.satservicegmbh.de
Tel +49 7738 99791-10

Table of Contents

Table of Contents 2
1 Overview 33
1.1 Using The Online Help 33
1.2 Architecture 34
2 M&C web-client 35
2.1 Top toolbar 36
2.1.1 Sidebar / main menu 36
2.1.2 History 37
2.1.3 Notification area 38
2.1.4 Theme 40
2.1.5 User menu 40
2.2 Main Content Window 40
2.2.1 User screens 41
2.2.2 Display elements 42
2.2.2.1 Text element 43
2.2.2.2 Frame element 44
2.2.2.3 Display element 44
2.2.2.4 Icon element 45
2.2.2.5 Arrow element 46
2.2.2.6 Rect element 46
2.2.2.7 Gauge element 47
2.2.2.8 Thumbnail element 47
2.2.2.9 Parameter element 47
2.2.2.10 Parameter button element 49
2.2.2.11 Latching button element 50
2.2.2.12 Radio button element 51
2.2.2.13 Button element 51
2.2.2.14 Device element 54
2.2.2.15 Switch element 54
2.2.2.16 Chart element 55
2.2.2.17 XY chart element 57
2.2.2.18 AzEl element 58
2.2.2.19 Spectrum element 60
2.2.2.20 Target list element 60
2.2.3 Device details panel 61
2.3 Device window 63
2.3.1 Device details panel 64
2.3.2 Device window 65
2.3.2.1 Device window pages 67
2.3.2.1.1 Fault page 67
2.3.2.1.2 Info page 68
2.3.2.1.3 Maintenance page 68
2.3.2.1.4 Config page 69
2.4 Treeview 70
2.5 Channel database 71
2.5.1 Satellite channels 72
2.5.1.1 Add/Edit satellite channel dialog 75

2.5.1.2 On delete confirmation dialog 78
2.5.2 IP channels 78
2.5.2.1 Add/Edit IP channel dialog** 81
2.5.2.2 On delete confirmation dialog 83
2.5.3 ASI channels 83
2.5.3.1 Add/edit ASI channel dialog** 86
2.5.3.2 On delete confirmation dialog 88
2.6 Device list 88
2.7 Device setup 90
2.7.1 Dialogs 92
2.8 File recorder 94
2.8.1 Functionality 96
2.8.2 Dialogs 97
2.9 Preset editor 98
2.9.1 Functionality 100
2.9.2 Dialogs 100
2.10 Inventory database 102
2.10.1 Inventory 102
2.10.1.1 Add/edit inventory dialog 104
2.10.2 Inventory related tables 106
2.10.2.1 Devices table 106
2.10.2.2 Add/edit device dialog 107
2.10.2.3 Add log dialog 108
2.10.2.4 Logs table 109
2.11 Documents 111
2.11.1 Upload/edit document 112
2.12 Satellite database 114
2.12.1 Satellite list 114
2.12.2 Add/edit satellite dialog 117
2.12.2.1 On delete confirmation dialog 118
2.12.3 Satellite related tables 118
2.12.3.1 I11 table 119
2.12.3.1.1 Functionality 119
2.12.3.1.2 Dialogs 120
2.12.3.2 Beacons table 121
2.12.3.2.1 Functionality 122
2.12.3.2.2 Dialogs 122
2.12.3.3 TC channel table 124
2.12.3.3.1 Functionality 125
2.12.3.3.2 Dialogs 126
2.12.3.4 TM channel table 128
2.12.3.4.1 Functionality 129
2.12.3.4.2 Dialogs 129
2.13 Satellite operator list 131
2.13.1 Functionality 133
2.13.2 Dialogs 133
2.14 Antenna list 135
2.14.1 Functionality 136
2.14.2 Dialogs 137

2.14.3 Antenna related tables 138
2.14.3.1 Satellite position table 138
2.14.3.1.1 Functionality 139
2.14.3.1.2 Dialogs 140
2.14.3.2 Tracks table 141
2.14.3.2.1 Functionality 142
2.14.3.2.2 Dialogs 142
2.14.3.3 Imports table 144
2.14.3.3.1 Functionality 145
2.14.3.3.2 Dialogs 145
2.15 The Macro Scheduler 146
2.15.1 Scheduler 147
2.15.2 Calender view 147
2.15.2.1 Features 149
2.15.3 Table view 149
2.15.3.1 Features 151
2.15.4 Unified dialogs 151
2.16 Screen editor 154
2.16.1 Functionality 157
2.16.2 Element properties 158
2.16.2.1 Display element types and their properties 160
2.16.2.1.1 Button element 161
2.16.2.1.2 Parameter element 163
2.16.2.1.3 Display element 165
2.16.2.1.4 Radio button element 167
2.16.2.1.5 Text element 168
2.16.2.1.6 Frame element 168
2.16.2.1.7 Rect element 169
2.16.2.1.8 Arrow element 170
2.16.2.1.9 Icon element 170
2.16.2.1.10 Device icon element 171
2.16.2.1.11 Parameter button element 173
2.16.2.1.12 Latching button element 175
2.16.2.1.13 Gauge element 176
2.16.2.1.14 Chart element 178
2.16.2.1.15 Spectrum element 179
2.16.2.1.16 Target list element 179
2.16.2.1.17 XY chart element 180
2.16.2.1.18 Thumbnail icon element 181
2.16.2.1.19 Switch element 182
2.16.2.1.20 AzEl element 184
2.17 Stream keys 184
2.17.1 Features 186
2.17.2 Dialogs 186
2.18 System status 187
2.18.1 Grid view 187
2.18.2 Table view 188
2.18.3 Common functions and details 188
2.18.4 Features 189

2.19 Treeview editor 190
2.19.1 Functionality 191
2.19.2 Dialogs 191
2.20 User management 194
2.20.1 Functionality 195
2.20.2 Dialogs 195
2.21 User settings 197
2.21.1 UI preferences 198
2.21.2 Warnings 198
2.21.3 Edit take mode 199
2.21.4 User screen 199
2.21.5 Navigation 200
2.21.6 Menu shortcuts 200
2.21.6.1 Functionality 201
2.21.6.2 Dialogs 201
2.21.7 Table settings 203
2.21.8 Reset 203
2.22 Global settings 203
2.23 Bottom panel 204
2.23.1 Interaction features 204
2.23.2 Live log 205
2.23.3 Event report 206
2.23.4 Active alarms 208
2.23.5 Macros 209
2.23.6 Terminal 210
3 Legacy Java Client Operation 212
3.1 The M&C Main Window 212
3.2 The M&C Tree View Window 213
3.3 The User Management Window 215
3.4 The Device Windows 216
3.5 The Terminal Window 218
3.6 The History File Viewer 220
3.7 Managing Device Presets 225
3.8 Recording and Playing Macros 227
3.8.1 The Macro Management window 227
3.8.2 The Macro Edit window 228
3.8.3 The Macro Record window 229
3.9 The Macro Scheduler 230
3.10 The Event Report Window 233
3.11 The Live Event Log Window 237
3.12 The Active Faults Window 239
3.13 CDB Central Database Table Editor 240
3.14 The sat-nms ACU Indoor Unit 243
3.14.1 Concepts 244
3.14.1.1 Tracking Modes 244
3.14.1.2 sat-nms ACU Indoor Unit & Outdoor Module Configuration 245
3.14.1.3 sat-nms ACU Indoor Unit & 3rd Party Controller Configuration 246
3.14.2 The ACU Main Window (with sat-nms ODM) 247
3.14.2.1 The Pointing Display 248

3.14.2.2 The Beacon Level Display 249
3.14.2.3 The Tracking Status Display 250
3.14.2.4 The ACU Subsystem Menu 251
3.14.2.5 The ODM Detailed Parameter Screens 252
3.14.2.6 The Satellite Select Screen 255
3.14.2.7 The Y/T Tracking History Viewer 257
3.14.3 The ACU Main Window (with 3rd party controller) 259
3.14.3.1 The Pointing Display 260
3.14.3.2 The Beacon Level Display 261
3.14.3.3 The Tracking Status Display 262
3.14.3.4 The ACU Subsystem Menu 263
3.15 Specialized Client Applications 264
3.15.1 Master Client 2 264
3.15.1.1 MasterClient 2 XML Configuration File 265
3.15.1.2 XML Element Description 266
3.15.1.3 Example showing a complete screenlist 270
3.15.2 Service Client Application 270
3.15.3 VioIP Client Application 274
3.15.4 VioIP Network Configuration Application 279
3.15.4.1 Groups 281
3.15.4.2 Profiles 283
3.15.4.3 Locations 284
3.15.4.4 Outgoing Lines 286
3.15.4.5 TX Devices 287
3.15.4.6 RX Devices 289
3.15.4.7 Adding New Devices 291
3.16 M&C Configuration 293
3.16.1 Configuring The Device Setup 293
3.16.1.1 Concepts 294
3.16.1.2 The Device Setup Editor Window 294
3.16.2 Arranging The User Interface 296
3.16.2.1 Concepts 296
3.16.2.2 The Screen Editor Window 298
3.16.2.3 The Element Properties Window 301
3.16.2.4 Display Element Types 303
3.16.2.4.1 Parameter 304
3.16.2.4.2 Read-Only Parameter 307
3.16.2.4.3 Fault 309
3.16.2.4.4 Label 310
3.16.2.4.5 3D Frame 310
3.16.2.4.6 Rectangle 311
3.16.2.4.7 Line / Arrow 313
3.16.2.4.8 Icon 315
3.16.2.4.9 EDIT Button 317
3.16.2.4.10 TAKE Button 318
3.16.2.4.11 Device Icon 318
3.16.2.4.12 Switch Icon 320
3.16.2.4.13 Frame Button 323
3.16.2.4.14 Parameter Button 327

3.16.2.4.15 Latching Button 330
3.16.2.4.16 Gauge (Bar Graph) 333
3.16.2.4.17 Strip Chart 335
3.16.2.4.18 Spectrum Display 336
3.16.2.4.19 MCP Display 337
3.16.2.4.20 Radio Buttons 338
3.16.2.4.21 LOCK Button 340
3.16.2.4.22 ODM Target List 342
3.16.2.4.23 XY Chart 343
3.16.2.4.24 JSON List 345
3.16.2.4.25 Thumbnail Icon 346
3.16.2.5 Configuring a Tabbed View Client 346
3.16.2.6 Configuring the M&C Tree View 349
4 Administration 351
4.1 Program Invocation 351
4.1.1 M&C Server 351
4.1.2 Backend Server 351
4.1.3 Client Programs 352
4.1.4 Automatic Software Update Of The Client 359
4.2 Configuration Files 361
4.2.1 backend.properties 361
4.2.1.1 Example configurations 365
4.2.2 user-ip-restrictions.json 366
4.2.3 vlc.properties 368
4.2.4 client.properties 374
4.2.5 satnms.rc 379
4.2.6 system.name 380
4.2.7 traprec.json 380
4.3 Event Database Configuration and Event Notification Add-Ons 383
4.3.1 Event To Serial Notification 384
4.3.2 Event To Mail Notification 385
4.3.3 Event To Script Notification 386
4.3.4 EDB.properties 387
4.4 Remote Control 388
4.4.1 SNMP Control 388
4.4.1.1 SNMP Parameter Translations 390
4.4.1.2 Configuring the SNMP Agent 392
4.4.1.3 The 'create-snmp.template' Utility Program 394
4.4.1.4 The 'create-satnms.mib' Utility Program 395
4.4.2 TCP/SSL Remote Control 395
5 Installation 398
5.1 Installation Base Debian Linux 398
5.1.1 Requirements 399
5.1.2 For the impatient 399
5.1.3 Installation 399
5.2 Installation MNC Service 403
5.2.1 Requirements 403
5.2.2 For the impatient 403
5.2.3 Installation 404

5.2.4 Configuration 405
5.3 Installation Backend Service 408
5.3.1 Requirements 408
5.3.2 For the impatient 409
5.3.3 Installation 409
5.3.4 Configuration 411
5.4 Installation SatDB database 414
5.4.1 Requirements 415
5.4.2 For the impatient 415
5.4.3 Installation 415
5.4.4 Configuration 417
5.4.5 PostgreSQL network configuration 419
5.4.5.1 Change users password 420
5.4.6 Manual configuration 420
5.4.6.1 Manual add database user 420
5.4.6.2 Create SatDB database 421
5.4.6.3 Load SatDB database schema 421
5.4.6.4 Force database version 422
5.4.6.5 Delete database user 422
5.4.6.6 Remove database 422
6 Theory Of Operation 424
6.1 M&C Concept 424
6.1.1 Parameter Messages 426
6.1.2 Device Drivers 427
6.1.3 Device Driver Variables 429
6.1.4 Device Driver Procedures 429
6.2 Event Logging 430
6.2.1 Concepts 430
6.3 Automating Procedures With Macros 431
6.3.1 Concepts 432
6.3.2 Macro Directives 433
6.3.3 Timing Issues 434
7 Device Driver Reference 436
7.1 Understanding the device driver 437
7.1.1 Variables 438
7.1.2 Procedures 439
7.1.3 Protocol encapsulation 439
7.2 Writing a device driver 440
7.2.1 Starting from scratch 440
7.2.1.1 General file format 441
7.2.1.2 Naming a device driver 441
7.2.1.3 Selecting a communication protocol 442
7.2.1.4 File includes 443
7.2.1.5 Tweaking the processing sequence 444
7.2.2 Defining variables 444
7.2.2.1 The VAR statement 445
7.2.2.2 The ALARM statement 450
7.2.2.3 Info variables 451
7.2.2.4 Setup variables 452

7.2.2.5 Variables for internal use 453
7.2.3 Using conversion tables 453
7.2.3.1 The TABLE statement 454
7.2.4 Adding data exchange procedures 454
7.2.4.1 The PROC statement 455
7.2.4.2 ASYNC procedures 457
7.2.5 Basic I/O functions 459
7.2.5.1 The PRINT statement 460
7.2.5.2 The INPUT statement 463
7.2.5.3 The WRITE statement 467
7.2.5.4 The READ statement 471
7.2.5.5 Function tables in I/O functions 475
7.2.6 REST I/O functions 476
7.2.6.1 The REST TRANSACT statement 479
7.2.6.2 The REST PARSE statement 480
7.2.6.3 The REST SET statement 482
7.2.6.4 The REST CLEAR statement 486
7.2.6.5 The REST SEND statement 486
7.2.7 Conditional execution 487
7.2.7.1 The IF statement 487
7.2.7.2 The GOTO statement 487
7.2.7.3 Label definition 488
7.2.8 Using subroutines 488
7.2.8.1 The CALL statement 489
7.2.9 Manipulating variables 489
7.2.9.1 The SET statement 490
7.2.9.2 The BITSET statement 490
7.2.9.3 The RANGESET statement 491
7.2.9.4 The BITSPLIT statement 495
7.2.9.5 The BITMERGE statement 495
7.2.9.6 The SEND statement 495
7.2.9.7 The COPY statement 496
7.2.10 More statements 496
7.2.10.1 The DELAY statement 497
7.2.10.2 The LOG statement 497
7.2.10.3 The DRATE statement 498
7.2.10.4 The SRATE statement 499
7.2.10.5 The WRITEHEX statement 500
7.2.10.6 The READHEX statement 501
7.2.10.7 The INVALIDATE statement 501
7.2.10.8 The SYNC statement 501
7.2.10.9 The RAISECOMMFAULT statement 502
7.3 The RPN language extension 502
7.3.1 The RPN stack 503
7.3.2 The { ...} statement 504
7.3.3 RPN command reference 505
7.4 SNMP device classification and variable binding 510
7.4.1 Device driver syntax for SNMP definitions 510
7.4.2 List of SNMP device classes and OID names 512

7.5 Device driver examples 515
7.5.1 NDSatCom-KuBand-Upconverter Example 515
7.5.2 Tandberg-Alteia Example 517
7.6 Device communication protocols 531
7.6.1 Writing a communication protocol definition 532
7.6.2 General file format 533
7.6.3 Global definitions 534
7.6.4 TX message elements 535
7.6.4.1 CHAR 536
7.6.4.2 ADDRESS 536
7.6.4.3 USERDATA 536
7.6.4.4 CHECKSUM 537
7.6.4.5 DATALENGTH 538
7.6.4.6 HEXLENGTH 538
7.6.4.7 SEQUENCE 538
7.6.5 RX message elements 539
7.6.5.1 START 539
7.6.5.2 CHAR 539
7.6.5.3 ADDRESS 540
7.6.5.4 USERDATA 540
7.6.5.5 STRING 541
7.6.5.6 CHECKSUM 541
7.6.5.7 DATALENGTH 543
7.6.5.8 HEXLENGTH 543
7.7 Device oriented user interface 543
7.7.1 How the software finds the screens for a device 544
7.7.2 Creating new screens 544
7.7.3 Creating a 'frame' definition 545
7.7.4 Icon reference 546
7.8 Online Help 549
7.8.1 Help file format 549
7.8.2 Rebuilding the online help 553
7.8.3 Adding help files for new devices 554
7.9 ServiceClient library 554
7.9.1 Adding new devices or device types to the library 555
7.9.2 Adapting the appearance of the ServiceClient 555
7.9.3 File format of definitions.xml 555
7.9.4 Device type definitions 555
7.9.4.1 Parameters 555
7.9.4.2 Implementations 555
7.9.5 GUI constants 555
7.10 Reference Tables 558
7.10.1 Device driver keyword reference 558
7.10.2 Protocol definition keyword reference 562
7.10.3 Help file keyword reference 563
7.10.4 Debugging with Terminal Session 564
7.10.4.1 Terminal Session Commands 564
7.10.4.2 Device Message Commands 569
7.10.5 Global faults 571

8 API Reference 573
8.1 HTTPS Communication and OAuth2 Authentication 573
8.2 API End Points 573
8.2.1 /token 574
8.2.2 /api/v1/logout 575
8.2.3 /healthcheck 576
8.2.4 /api/v1/user 576
8.2.5 /api/v1/users 578
8.2.6 /api/v1/mnclist 579
8.2.7 /api/v1/activebackend 579
8.2.8 /api/v1/peek 580
8.2.9 /api/v1/poke 581
8.2.10 /api/v1/makewebsocket 582
8.2.11 /api/v1/subscribe 582
8.2.12 /api/v1/uscreens 585
8.2.13 /api/v1/dscreens 588
8.2.14 /api/v1/dframes 590
8.2.15 /api/v1/macros 593
8.2.16 /api/v1/playmacro 595
8.2.17 /api/v1/presets 596
8.2.18 /api/v1/applypreset 599
8.2.19 /api/v1/collectpreset 601
8.2.20 /api/v1/presetvars 601
8.2.21 /api/v1/devicevars 602
8.2.22 /api/v1/schedule 603
8.2.23 /api/v1/healthcheck 604
8.2.24 /api/v1/dbstate 605
8.2.25 /api/v1/dbversion 605
8.2.26 /api/v1/satellites 606
8.2.27 /api/v1/satoperators 630
8.2.28 /api/v1/channels 633
8.2.29 /api/v1/streamkeys 639
8.2.30 /api/v1/antennas 642
8.2.31 /api/v1/satdetails 663
8.2.32 /api/v1/tleimport 664
8.2.33 /api/v1/eventreport 664
8.2.34 /api/v1/eventack 665
8.2.35 /api/v1/trackmem 666
8.2.36 /api/v1/treeview 667
8.2.37 /api/v1/debug 669
8.2.38 /api/v1/redundancy 670
8.2.39 /api/v1/restart 672
8.2.40 /api/v1/devicesetup 673
8.2.41 /api/v1/drivers 674
8.2.42 /api/v1/protocols 675
8.2.43 /api/v1/preferredprotocols 675
8.2.44 /api/v1/filerecordings 676
8.2.45 /api/v1/filerecordersettings 677
8.2.46 /api/v1/acutargets 679

8.2.47 /api/v1/inventory 679
8.2.48 /api/v1/documents 688
8.2.49 /api/v1/thumbnail 693
8.2.50 /api/v1/backendinfo 693
8.2.51 /api/v1/globalSettings 694
8.2.52 /api/v1/userSettings 695
8.3 WebSocket Communication 698
8.3.1 Creating WebSockets 698
8.3.2 Subscribing For Messages 698
8.3.3 Interpreting WebSocket Data 699
8.3.4 Echo / Heartbeat Function 699
8.3.5 Closing Websockets 699
8.4 Debug Terminal Communication 700
8.4.1 Opening A Debug Terminal 700
8.4.2 Closing A Debug Terminal 700
8.4.3 Sending Debug Commands 700
8.4.4 M&C Redundancy Management 700
8.4.4.1 Redundancy Commands 701
8.5 Data Models 701
8.5.1 TokenReply 704
8.5.2 UserProperties 704
8.5.3 UserList 705
8.5.4 ApiError 705
8.5.5 MncList 706
8.5.6 MncInfo 706
8.5.7 ActiveBackend 706
8.5.8 SubscribeList 707
8.5.9 WebsocketId 707
8.5.10 Message 707
8.5.11 Parameter 709
8.5.12 Range 710
8.5.13 StringRange 711
8.5.14 IntegerRange 712
8.5.15 HexRange 712
8.5.16 DoubleRange 713
8.5.17 BooleanRange 714
8.5.18 AlarmFlagRange 714
8.5.19 EnumRange 715
8.5.20 ObjectRange 715
8.5.21 SpectrumTrace 716
8.5.22 TrackingPoint 716
8.5.23 TrackingHistory 717
8.5.24 EventLog 717
8.5.25 EventReport 718
8.5.26 EventReportRequest 718
8.5.27 EventAck 719
8.5.28 EventInfo 720
8.5.29 FaultList 721
8.5.30 FaultInfo 721

8.5.31 DeviceFrameDefinition 721
8.5.32 DeviceFrameTab 722
8.5.33 ScreenDefinition 722
8.5.34 ScreenElement 723
8.5.35 ArrowElement 725
8.5.36 ButtonElement 727
8.5.37 ChartElement 731
8.5.38 AzElElement 733
8.5.39 DeviceElement 734
8.5.40 DisplayElement 736
8.5.41 EditButtonElement 737
8.5.42 FaultElement 738
8.5.43 FrameElement 738
8.5.44 GaugeElement 738
8.5.45 IconElement 741
8.5.46 LatchingButtonElement 742
8.5.47 LockButtonElement 744
8.5.48 MCPElement 744
8.5.49 ParameterButtonElement 744
8.5.50 ParameterElement 746
8.5.51 RadioButtonElement 748
8.5.52 RectElement 749
8.5.53 SpectrumElement 751
8.5.54 SwitchElement 752
8.5.55 TakeButtonElement 754
8.5.56 TargetListElement 754
8.5.57 TextElement 755
8.5.58 ThumbnailElement 756
8.5.59 XYChartElement 757
8.5.60 ColorDefinition 758
8.5.61 ItemList 758
8.5.62 Macro 759
8.5.63 PresetValue 759
8.5.64 DevicePreset 759
8.5.65 PresetVars 760
8.5.66 DeviceVars 760
8.5.67 KeyValPair 761
8.5.68 Dictionary 761
8.5.69 SatDbSatellite 761
8.5.70 SatDbBeaconAttenuation 762
8.5.71 SatDbBeacon 762
8.5.72 SatDbTcAttenuation 763
8.5.73 SatDbTc 763
8.5.74 SatDbTmAttenuation 764
8.5.75 SatDbTm 764
8.5.76 SatDbSatOperator 765
8.5.77 SatDbPosition 766
8.5.78 SatDbAntenna 767
8.5.79 SatDbTLEData 767

8.5.80 SatDbI11Data 768
8.5.81 SatDbTableTracking 769
8.5.82 SatDbSatDetails 769
8.5.83 SatDbTleImport 770
8.5.84 SatDbState 770
8.5.85 ChannelData 771
8.5.86 StreamKeyData 774
8.5.87 TreeViewNode 774
8.5.88 DatabaseVersion 777
8.5.89 DebugMessage 777
8.5.90 RedundancyState 777
8.5.91 RedundancySummary 778
8.5.92 RedundancyCmd 778
8.5.93 RestartCmd 778
8.5.94 ScheduleEvent 779
8.5.95 Schedule 780
8.5.96 DeviceDefinition 780
8.5.97 DeviceThreadDefinition 780
8.5.98 DeviceSetup 780
8.5.99 FRViewProperties 781
8.5.100 FRViewTraceDescription 781
8.5.101 FRViewPreset 781
8.5.102 FRViewTraceScaling 782
8.5.103 AcuTarget 782
8.5.104 AcuTargetList 783
8.5.105 InventoryItem 783
8.5.106 InventoryDevice 783
8.5.107 InventoryLogEntry 784
8.5.108 DocumentItem 784
8.5.109 DocumentList 785
8.5.110 Thumbnail 785
8.5.111 BackendInfo 785
8.5.112 UISettings 786
8.6 Satellite Database 786
8.6.1 Table Satellites (satellites) 788
8.6.2 Table Satellite Operators (satellites_operators) 788
8.6.3 Antennas (antennas) 788
8.6.4 Table Satellite Beacons (satellite_beacons) 789
8.6.5 Table Satellite TC (satellite_tc) 789
8.6.6 Table Satellite TM (satellite_tm) 790
8.6.7 Table Satellite Positions (satellite_positions) 790
8.6.8 Table Table Tracking (table_tracking) 791
8.6.9 Table Intelsat 11 Parameters (intelsat_11parameters) 791
8.6.10 Table Norad TLE (norad_tle) 792
8.6.11 Table Channels (channels) 792
8.6.12 Table StreamKeys (stream_keys) 795
8.6.13 Table BeaconAttenuations (beacon_attenuations) 795
8.6.14 Table TcAttenuations (tc_attenuations) 796
8.6.15 Table TmAttenuations (tm_attenuations) 796

8.6.16 Table InventoryItems (inventory_items) 796
8.6.17 Table InventoryDevices (inventory_devices) 797
8.6.18 Table InventoryLog (inventory_log) 797
8.6.19 Table Documents (documents) 798
8.6.20 Table GlobalSettings (globalSettings) 798
8.6.21 Table User Settings (userSettings) 799
8.7 Fonts 799
9 Device Driver Descriptions 801
9.1 Device Drivers By Name 801
9.2 Device Drivers By Vendor 854
9.3 Device Drivers By Tag 906
9.4 Logical Devices Overview 967
9.5 ACTi-ACM1231-Webcam 969
9.6 APC-ATS 970
9.7 APC-RPDU 974
9.8 APS-Backup-Antenna-Control 978
9.9 APS-Backup-Antenna-Switch 982
9.10 APS-Site-Switch 986
9.11 ARD-Backup-Client 990
9.12 ARD-Backup-Server 993
9.13 ASCSignal-NGC-ACU 998
9.14 ASCSignal-NGC-BDC 1002
9.15 ASCSignal-NGC-Deice 1005
9.16 ASCSignal-NGC-Hub-AC 1011
9.17 ASCSignal-NGC-RedundancyModule 1016
9.18 ASCSignal-NGC-Temperature 1022
9.19 ASCSignal-NGC-Weather 1028
9.20 Acorde-ACRS-IDU 1030
9.21 Acorde-BUC-ACCVT 1035
9.22 Actia-Sodielec-DNT-HPA 1039
9.23 Actia-Sodielec-DSS1300 1042
9.24 Adtec-Afiniti-Decoder 1045
9.25 Adtec-Afinity-Encoder 1050
9.26 Adtec-EN81 1062
9.27 Adtec-RD60 1069
9.28 Advantech-ADAM5K 1075
9.29 Advantech-ADAM6K 1079
9.30 Advantech-Genesis 1081
9.31 Advantech-SSPB-GCP 1086
9.32 Advantech-SSPB-MCP 1090
9.33 Advantech-SSPB-SNMP 1093
9.34 Advantech-SSPB 1095
9.35 Agilent-Spectrum-Analyzer 1097
9.36 Agilis-BUC 1103
9.37 Altec-Puregas-P210 1106
9.38 Altec-Puregas-P550 1110
9.39 Anacom-KuBand-Transceiver 1113
9.40 Anacom-Protection-Switch 1119
9.41 Andrew-APC100 1121

9.42 Antech-Deice 1124
9.43 Antenna-BBU-Connector 1126
9.44 Antenna-Dummy 1129
9.45 Antenna-Management 1130
9.46 Antenna-Pointing-Fixed 1144
9.47 Antenna-Pointing-Hiltron 1149
9.48 Antenna-Pointing-ODM 1151
9.49 Antenna-Pointing 1153
9.50 Antenna-Proxy 1158
9.51 Antenna-Tracking 1159
9.52 Antenna-Wind-Protection 1169
9.53 AntwerpSpace-G3-Demod 1174
9.54 AntwerpSpace-G3-Mod 1184
9.55 AntwerpSpace-IBB-Modem 1189
9.56 Apollo-Switch 1201
9.57 Arithmetic 1204
9.58 Astro-EdgeStreamer 1206
9.59 Ateme-DR5000 1245
9.60 Ateme-DR8400 1261
9.61 Ateme-Kyrion-CM5000 1274
9.62 Ateme-TitanEdge-Encoder 1292
9.63 Ateme-TitanEdge-Gateway 1320
9.64 Ateme-TitanEdge-IRD 1351
9.65 Aten-VGA-Splitter 1375
9.66 AtlanTecRF-TLT 1378
9.67 AtlanticSat-ASC300 1380
9.68 Auto-RollOff 1382
9.69 AutomationDirect-Terminator 1384
9.70 AvL-ITC-ACU 1390
9.71 AvL-TracStar 1393
9.72 AzElCollector 1405
9.73 BISS-KeySelect-CDB-DBonly 1407
9.74 BISS-KeySelect-CDB 1409
9.75 BISS-KeySelect 1412
9.76 Band-Pol-4-To-1 1414
9.77 Barco-RE4220 1416
9.78 Bard-MC4002-AC 1422
9.79 Blankom-DRP393-SNMP 1425
9.80 Brainbox-EDxxx 1432
9.81 CGC-ACU 1435
9.82 CPI-350W-TWTA 1445
9.83 CPI-CHPA 1449
9.84 CPI-GaN-SSPA 1454
9.85 CPI-KHPA-G3 1468
9.86 CPI-KHPA-G4 1472
9.87 CPI-LPA 1478
9.88 CPI-Mini-ODU 1482
9.89 CPI-SLPA 1496
9.90 CPI-SRCU-BCIP 1498

9.91 CPI-SRCU 1502
9.92 CPI-Switch 1505
9.93 CPI-TWTA-Ka 1507
9.94 CPI-TWTA 1512
9.95 CPI-TouchPower 1516
9.96 CPI-VZU-TWTA 1529
9.97 CSM-Spectrum-Analyzer 1531
9.98 Calian-Decimator 1545
9.99 Calian-MCI-ACU 1555
9.100 CandM-Switch 1564
9.101 Carrier-Site-Switch 1567
9.102 Channel-Select 1574
9.103 ChinaSuncom-9293S3-ACU 1575
9.104 ChinaSuncom-Beacon-Receiver-SC9800 1579
9.105 Choice-Translator 1582
9.106 Choice 1584
9.107 Cibred-LAB4 1585
9.108 Cisco-2800GSM-SNMP 1586
9.109 Cisco-2900-SNMP 1588
9.110 Cisco-D9800-IRD 1592
9.111 Cisco-Monitor 1599
9.112 Cisco-Router 1603
9.113 Codan-7586L 1605
9.114 Codan-BUC-7586L 1608
9.115 Codan-LBUC6x00 1612
9.116 Codan-Transceiver 1615
9.117 Collector-Two-Way 1619
9.118 Collector 1622
9.119 ColumbiaWeather-MicroServer 1627
9.120 ComWinTop-THxxS 1631
9.121 Comment 1632
9.122 Compare 1633
9.123 Comstream-ABR200 1635
9.124 Comstream-ABR202 1642
9.125 Comstream-CM601A-SCPC 1649
9.126 Comstream-CM701-MC 1655
9.127 Comstream-CM701 1664
9.128 Comstream-CX101 1673
9.129 Comstream-CX801 1675
9.130 Comstream-DAC700-PRE 1678
9.131 Comstream-DAC700 1687
9.132 Comstream-Demodulator 1692
9.133 Comstream-LoSpeed-Modem-SCPC 1696
9.134 Comstream-Modem-Card-SCPC 1702
9.135 Comstream-Modulator 1709
9.136 Comtech-CDD564 1713
9.137 Comtech-CDM570 1725
9.138 Comtech-CDM600 1733
9.139 Comtech-CDM760 1744

9.140 Comtech-DT4500-SNMP 1758
9.141 Comtech-DT4500 1765
9.142 Comtech-LBC4000 1768
9.143 Comtech-LPOD-SNMP 1773
9.144 Comtech-MBT5000 1776
9.145 Comtech-MBT5000A 1780
9.146 Comtech-UT4500-SNMP 1786
9.147 Comtech-UT4500 1793
9.148 Conditional-Replicator 1796
9.149 Cross-Frequency-Source 1798
9.150 Cross-UpDownConverter 1800
9.151 DEV-4111-DA 1802
9.152 DEV-Distribution-2190 1806
9.153 DEV-FO-RX 1815
9.154 DEV-FO-TX 1818
9.155 DEV-Frame 1822
9.156 DEV-Matrix-1986 1828
9.157 DEV-Matrix-1996 1860
9.158 DEV-Switch-2-To-1 1874
9.159 Dantherm-ComboCooling1000 1877
9.160 Dantherm-TKS3000A 1880
9.161 De-Icing-Buttons 1882
9.162 De-Icing-Multiplex 1884
9.163 Dektec-DTA2137 1888
9.164 Dektec-DTA2137v4 1893
9.165 Delay-Line-Mirror 1897
9.166 Delay-Line 1899
9.167 Delayed-Gate-3 1901
9.168 Device-Copy 1904
9.169 Device-Mirror 1905
9.170 Device-Switcher 1907
9.171 DoubleD-DDA89 1909
9.172 Downconverter-Protection-Controller 1914
9.173 Dual-Timezone-Clock 1918
9.174 Dual-Translator 1919
9.175 Dual-Wind-Speed 1921
9.176 E2V-TWTA 1923
9.177 ECU-J0475 1928
9.178 EF-Data-SDM300 1931
9.179 ET-ADH-2A 1939
9.180 ETI-ADH-NETCOM 1942
9.181 ETL-2773-XXYY 1944
9.182 ETL-Alto-Amp 1950
9.183 ETL-Alto-DA 1956
9.184 ETL-Dextra-Combiner 1965
9.185 ETL-Dextra-DistAmp 1968
9.186 ETL-Genus-100MHz 1978
9.187 ETL-Genus-10MHz 1982
9.188 ETL-Genus-ALT 1988

9.189 ETL-Genus-COM402 1993
9.190 ETL-Genus-DA-22512 1996
9.191 ETL-Genus-DIV401 2006
9.192 ETL-Genus-Downconverter 2010
9.193 ETL-Genus-Frame 2015
9.194 ETL-Genus-PRN 2021
9.195 ETL-Genus-SRY-RX 2025
9.196 ETL-Genus-SRY-TX 2028
9.197 ETL-Genus-Upconverter 2033
9.198 ETL-Genus-VTR-101-Matrix 2038
9.199 ETL-Genus-VTRC-101-Matrix 2048
9.200 ETL-HMI-Reboot 2058
9.201 ETL-Hurricane-Matrix 2059
9.202 ETL-LS-Switch 2087
9.203 ETL-MatrixPort-Emulator 2091
9.204 ETL-Modular-Frame-26128 2092
9.205 ETL-NGM100-Matrix 2095
9.206 ETL-NGM100C-Matrix 2135
9.207 ETL-NGM21C-Matrix 2164
9.208 ETL-NGM30-Matrix 2191
9.209 ETL-NSN100-Matrix 2220
9.210 ETL-PDSwitch-Emulator 2363
9.211 ETL-POE-Switch 2366
9.212 ETL-Piranha 2368
9.213 ETL-SHF-Switch-2to2 2393
9.214 ETL-SHF-Switch 2395
9.215 ETL-Splitter-DIV409 2397
9.216 ETL-Splitter-DIV429 2399
9.217 ETL-Stingray-FO-Base 2402
9.218 ETL-Stingray-FO-RX 2405
9.219 ETL-Stingray-FO-TX 2409
9.220 ETL-SwitchUnit-23258 2413
9.221 ETL-VTR70-Matrix 2417
9.222 ETL-VTR70C-Matrix 2429
9.223 ETL-VTX100-Matrix 2440
9.224 ETL-Volcano-Matrix 2458
9.225 Emerson-AC-HPM 2838
9.226 Endrun-FDC3300 2841
9.227 Enensys-Vyper-Modulator 2843
9.228 Envivio-4Caster-B3 2867
9.229 Envivio-4View-D1 2876
9.230 Ericsson-AVP 2879
9.231 Ericsson-Voyager-II-v9.5 2903
9.232 Ericsson-Voyager-II 2928
9.233 Evertz-7700-Frame 2952
9.234 Evertz-7703-BPX 2955
9.235 Evertz-7707-ET 2958
9.236 Evertz-7707-LR 2960
9.237 Evertz-7707-LT 2963

9.238 Evertz-7708-LR 2965
9.239 Evertz-7708-LT 2968
9.240 Evertz-7800-Frame 2971
9.241 Evertz-7881-IRD 2974
9.242 Evertz-EMR-Matrix 2983
9.243 Execute 3008
9.244 Experimental 3010
9.245 Fanuc-PLC9030 3011
9.246 Fault-Array 3013
9.247 Fault-Gate-3 3016
9.248 Fault-Gate 3018
9.249 FeiZyfer-GPS-Receiver 3020
9.250 File-Recorder 3022
9.251 Filter 3024
9.252 Floating-Limit 3025
9.253 Foxcom-SatLight-Platinum 3027
9.254 Frequency-Loop 3042
9.255 Frequency-Split 3044
9.256 GDSatcom-ACU930 3046
9.257 GDSatcom-ACU940 3057
9.258 GDSatcom-ACU950 3070
9.259 GDSatcom-ACU970 3084
9.260 GDSatcom-ACU980 3099
9.261 GDSatcom-DTR520 3120
9.262 GDSatcom-RSC1200 3123
9.263 GDSatcom-Rack-SSPA 3126
9.264 GDSatcom-SSPA-QuadMod 3128
9.265 Gate-Array 3135
9.266 Generex-CS121-UPS-Adapter 3139
9.267 Generic 3142
9.268 Geosync-PS 3142
9.269 Geosync-RCU-R2R 3146
9.270 Geosync-RCU 3148
9.271 Geosync-TLT 3149
9.272 Gude-ExpertPowerControl 3152
9.273 GundD-CompactCenter 3156
9.274 HP-Powermeter-Legacy 3158
9.275 HP-Powermeter-SCPI 3160
9.276 HP-Synthesizer-SCPI 3162
9.277 HWGroup-Poseidon2 3164
9.278 HWGroup-STE 3173
9.279 HWGroup-STE2 3176
9.280 Haivision-Makito-X-Decoder 3177
9.281 Haivision-Makito-X 3183
9.282 Haivision-SRT-Gateway 3196
9.283 Harris-Panacea 3213
9.284 Hilkom-BNova-SNMP 3221
9.285 Hiltron-ACU2 3231
9.286 Hiltron-Deice 3243

9.287 Hiltron-HCS-Frame 3246
9.288 Hiltron-HCS-PS 3247
9.289 Hiltron-HCS-Power 3249
9.290 Hiltron-HCS-SSPA 3251
9.291 Hiltron-HCS-Switch 3254
9.292 Hiltron-HCS-TWTA 3256
9.293 Hiltron-Multi-Deice 3260
9.294 Hitec-AFP-ACU 3263
9.295 Hitec-AFP-Deice 3264
9.296 Hitec-HACU-01-NAOS 3267
9.297 Hitec-HACU-01-V3.02 3284
9.298 Hitec-HACU-01-V3 3292
9.299 Hitec-HACU-01 3310
9.300 Hitec-HAFP-00-NAOS1 3317
9.301 Hitec-HAFP-00 3321
9.302 Hitec-HSCU-00 3323
9.303 IDirect-5000 3339
9.304 IP-Sens-IPS2225P 3343
9.305 IRD-Chain 3349
9.306 Infratec-PM 3354
9.307 Inmarsat-NLES-Status 3357
9.308 Input 3360
9.309 Irt-Pico-BUC-RCP 3364
9.310 Janitza-UMG604 3367
9.311 KBR-Multimess-4F96 3369
9.312 Key-Generator 3371
9.313 Keysight-FieldFox 3372
9.314 LPT-3000RX4 3381
9.315 Leitch-Panacea 3390
9.316 Level-Add 3392
9.317 Level-AddRed 3394
9.318 Level-Control 3398
9.319 Level-Monitoring 3408
9.320 Level-Set 3410
9.321 LineUp-Control 3415
9.322 Link-Mirror 3417
9.323 Link-Static 3419
9.324 Link-Two-Way 3420
9.325 Link 3425
9.326 Logger 3429
9.327 Logical-Gate-3 3431
9.328 Logical-Gate 3434
9.329 Lynx-DVD5420-ASI-Distr 3436
9.330 Lynx-DVD5820-ASI-Distr 3440
9.331 Lynx-DVD5820-Main 3442
9.332 MCL-MT-TWTA-CSP 3444
9.333 MCL-MT-TWTA 3451
9.334 MFM-IpDecoder 3456
9.335 MFM-IpEncoder 3460

9.336 MFM-Ird 3466
9.337 MFM-PlayList 3472
9.338 MFM-SkyWANIdu 3480
9.339 MFM-Upc 3483
9.340 MIB2-Ethernet-Switch 3491
9.341 MIB2-Interface-Monitor 3533
9.342 MIB2-Storage-Monitor 3542
9.343 MIB2-UPS-Monitor 3553
9.344 Macro-Player 3556
9.345 Matrix-Protection-Switcher 3558
9.346 Matrix-Site-Switch 3561
9.347 Maxtech-LRC-1200 3565
9.348 Measurement-Converter 3567
9.349 Mediakind-CE1 3569
9.350 Mediakind-RX1 3585
9.351 Meinberg-LantimeM300 3604
9.352 Meinberg-SDU 3607
9.353 Mesa-WNT 3610
9.354 Mesa-WSWD 3611
9.355 MiniCircuits-Switch-ZT 3613
9.356 Mirad-VPC 3616
9.357 MissionMicrowave-SSPA 3618
9.358 Mitec-ARK-RX 3620
9.359 Mitec-Switch 3622
9.360 Mitec-TX 3624
9.361 Miteq-DVM100 3627
9.362 Miteq-Downconverter-9900-SNMP 3632
9.363 Miteq-Downconverter-9900 3635
9.364 Miteq-KBand-Upconverter 3638
9.365 Miteq-KuBand-Upconverter 3639
9.366 Miteq-LBand-Downconverter 3641
9.367 Miteq-LBand-Upconverter 3643
9.368 Miteq-NSU 3645
9.369 Miteq-TLT-LO 3653
9.370 Miteq-TLT 3654
9.371 Miteq-UPC 3657
9.372 Miteq-Upconverter-9900-SNMP 3660
9.373 Modcod-Splitter 3664
9.374 Modem-Protection-Controller 3665
9.375 Monitor-Switch 3669
9.376 Monterey-140 3671
9.377 Moxa-ioLogik-1200 3673
9.378 Moxa-ioLogik-2214 3675
9.379 Moxa-ioLogik-4000 3678
9.380 Moxa-ioLogik-4200 3685
9.381 Moxa-ioLogik-Current 3692
9.382 Moxa-ioLogik-Temperature 3693
9.383 Moxa-ioThinx-4510 3695
9.384 Multi-Add 3700

9.385 Multi-Protection-Switch 3705
9.386 Multi-Site-Switch 3728
9.387 Multi-Symmetric-Site-Switch 3737
9.388 N-Way-Switch-Latching 3747
9.389 N-Way-Switch 3751
9.390 NAIS-FP-8-Port-Switch 3753
9.391 NAIS-FP-AD-Test 3756
9.392 NAIS-FP-Alarm-Circuit 3756
9.393 NAIS-FP-Inv-Alarm-Circuit 3758
9.394 NAIS-FP-Inv-Pulse-Output 3760
9.395 NAIS-FP-Latching-Relay 3762
9.396 NAIS-FP-Protection-Switch 3765
9.397 NAIS-FP-Pulse-Output 3766
9.398 NAIS-FP-RTD-Module 3768
9.399 NAIS-FP-Relay 3772
9.400 NAIS-FP-Simple-Input 3774
9.401 NAIS-FP-Simple-Output 3776
9.402 NAIS-FP-TxOn-Switch 3777
9.403 NAIS-FP-Watchdog-Output 3779
9.404 NAIS-FP0-REG 3780
9.405 NAIS-FP0-RO 3783
9.406 NAIS-FP0 3785
9.407 NDSatCom-1Axis-ACU 3788
9.408 NDSatCom-3Axis-ACU 3791
9.409 NDSatCom-ACU4100A-SNG 3794
9.410 NDSatCom-ACU4100B 3797
9.411 NDSatCom-ACU5000-SNG 3800
9.412 NDSatCom-ACU5030 3808
9.413 NDSatCom-ACU6030-REST 3811
9.414 NDSatCom-CPI 3814
9.415 NDSatCom-KBand-Downconverter 3818
9.416 NDSatCom-KBand-Upconverter 3821
9.417 NDSatCom-KuBand-Downconverter 3823
9.418 NDSatCom-KuBand-Upconverter 3825
9.419 NDSatCom-KuHBand-Downconverter 3827
9.420 NDSatCom-KuHBand-Upconverter 3829
9.421 NDSatCom-KuLBand-Downconverter 3831
9.422 NDSatCom-KuLBand-Upconverter 3833
9.423 NDSatCom-RCU-RC11 3835
9.424 NDSatCom-RCU-RC11OMa 3838
9.425 NDSatCom-RCU-SNMP 3841
9.426 NDSatCom-SkyWAN-IDU 3845
9.427 NDSatCom-Step-Track 3849
9.428 NDSatCom-TWTA-BC30 3851
9.429 NDSatCom-TWTA-REST 3855
9.430 NDSatCom-TWTA-SNMP 3859
9.431 NDSatCom-TWTA 3863
9.432 NMEA-GPS-Receiver 3868
9.433 NPC-ACU-550 3869

9.434 NTT-HVD6100 3877
9.435 NTT-HVE9100 3885
9.436 NTT-HVE9210 3891
9.437 Narda-MCP 3900
9.438 Narda-NRA 3907
9.439 Narda-Upconverter 3916
9.440 Negate 3918
9.441 Netgear-M4300-Monitor 3920
9.442 Nevion-Sublime 3940
9.443 Nevion-TVG420 3942
9.444 Nevion-TVG425 3963
9.445 Newtec-AZ110 3983
9.446 Newtec-AZ200-1-To-1 3990
9.447 Newtec-AZ410 3993
9.448 Newtec-AZ910 4014
9.449 Newtec-EL470 4024
9.450 Newtec-FRC0710 4029
9.451 Newtec-FRC0750 4033
9.452 Newtec-M6100 4037
9.453 Newtec-MCX7000-ASIIN 4053
9.454 Newtec-MCX7000-ASIOUT 4055
9.455 Newtec-MCX7000-BISS-Descr 4057
9.456 Newtec-MCX7000-Base 4059
9.457 Newtec-MCX7000-Demodulator 4062
9.458 Newtec-MCX7000-IPIN 4067
9.459 Newtec-MCX7000-IPOUT 4070
9.460 Newtec-MCX7000-Modulator 4072
9.461 Newtec-MCX7000-Protection 4077
9.462 Newtec-MDM6100 4079
9.463 Newtec-NTC2080 4094
9.464 Newtec-NTC2142 4099
9.465 Newtec-NTC2180 4103
9.466 Newtec-NTC2277 4109
9.467 Newtec-PEP 4116
9.468 Newtec-USS0202 4117
9.469 Newtec-USS0212 4120
9.470 Noise-Source-Measurement 4123
9.471 NovelSat-NS2000 4125
9.472 Novella-Blockdownconverter 4131
9.473 Novella-Blockupconverter 4133
9.474 Novella-RU03 4135
9.475 Novella-RU04 4137
9.476 Novella-Tracking-Receiver 4140
9.477 Novotronik-GTA5200 4141
9.478 Novotronik-GTS0118 4143
9.479 Novotronik-GTS0752 4146
9.480 Novotronik-GTS0752A 4148
9.481 Novotronik-GTS4600 4150
9.482 Novotronik-GTS5022 4151

9.483 Novotronik-GTS5460 4153
9.484 Novotronik-GTS5852 4155
9.485 Novotronik-GTS6300 4157
9.486 Novotronik-GTS6303 4159
9.487 Novotronik-GTS6305 4163
9.488 Novotronik-GTS6821 4165
9.489 Novotronik-GTS6832 4168
9.490 Novotronik-GTV2455 4170
9.491 Novotronik-MAS3160 4172
9.492 Novotronik-MAS3660 4175
9.493 Novotronik-MAS3670 4194
9.494 Novotronik-MAS5321 4205
9.495 Novotronik-MAS5513 4210
9.496 Novotronik-MAS5514 4214
9.497 Novotronik-Matrix24 4217
9.498 Novotronik-Matrix32 4221
9.499 Novotronik-Matrix8 4227
9.500 Novus-ND2310D 4229
9.501 Novus-NR2110-SNMP 4235
9.502 Novus-NR2110 4239
9.503 Number-Select 4241
9.504 Number-Set 4243
9.505 ORB-Step-Track 4245
9.506 Off-Switch 4247
9.507 OnTarget 4248
9.508 OpenGear-Frame 4250
9.509 Output 4252
9.510 Paradise-RCP2-1100-SNMP 4256
9.511 Paradise-RCP2-SNMP 4259
9.512 Paradise-RCP2 4261
9.513 Paradise-SSPA-SNMP 4263
9.514 Parameter-Mirror 4266
9.515 Pass-Scheduler 4268
9.516 Peak-Beacon-Receiver 4276
9.517 Peak-Blockdownconverter 4279
9.518 Peak-Blockupconverter 4281
9.519 Peak-DBU200 4284
9.520 Peak-DBU200H 4286
9.521 Peak-Downconverter-7000 4289
9.522 Peak-IBDH 4291
9.523 Peak-P700 4293
9.524 Peak-UPC7002 4296
9.525 Peak-Upconverter-7000 4301
9.526 Pico-EnviroMon-EL005 4304
9.527 Pointing-Lock 4309
9.528 Pointing-Robot 4312
9.529 Polarization-Select 4318
9.530 PowerCalc-Transponder 4319
9.531 PowerCalc 4321

9.532 Profline-Plus1-AudioReceiver 4323
9.533 Promax-Sathunter 4333
9.534 Protection-Switch-1-To-1-Dual 4336
9.535 Protection-Switch-1-To-1 4341
9.536 Protection-Switch-2-To-1 4346
9.537 Protection-Switch-Asymmetric 4350
9.538 Protection-Switch-N-To-1 4355
9.539 Protection-Switch-N-To-M 4369
9.540 Protection-Switch-NoSave-1-To-1 4371
9.541 Proxy-Device 4376
9.542 Qbit-Q572-AudioReceiver 4379
9.543 RFDesign-Amp 4387
9.544 RFDesign-FLCR-ODA 4389
9.545 RFDesign-FLCR 4403
9.546 RFDesign-FlexLink-S7B 4429
9.547 RFDesign-FlexLink-S7Bc 4443
9.548 RFDesign-LSC 4454
9.549 RFDesign-QLink 4478
9.550 RFDesign-RLC 4486
9.551 RFDesign-RLS 4489
9.552 RFDesign-RSC 4493
9.553 RFDesign-RedLink 4495
9.554 RX-Channel-ActivationBased 4520
9.555 RX-Channel-ByChannel 4528
9.556 RX-Channel-BySatellite 4537
9.557 RX-Channel 4545
9.558 RX-ChannelSelect-CDB-DBonly 4561
9.559 RX-ChannelSelect-CDB 4565
9.560 RX-ChannelSelect 4574
9.561 RX-LineSettings-CDB-DBonly 4582
9.562 RX-LineSettings-CDB 4588
9.563 RadeusLab-ACU-8200 4598
9.564 Radyne-DM240 4610
9.565 Radyne-DMD2050 4613
9.566 Radyne-DVB3030 4625
9.567 Radyne-RCS11 4629
9.568 Radyne-SFC1450 4630
9.569 Raritan-EMX 4632
9.570 Remote-Alarm 4648
9.571 Replicator 4651
9.572 Report-Generator 4653
9.573 ResearchConcepts-RC1500 4658
9.574 ResearchConcepts-RC2000 4662
9.575 ResearchConcepts-RC2500 4667
9.576 ResearchConcepts-RC3000 4673
9.577 Reset-Timer 4679
9.578 Romantis-UHP 4681
9.579 RuS-FPH 4684
9.580 RuS-FSC 4693

9.581 RuS-FSH 4702
9.582 RuS-FSL 4711
9.583 RuS-FSV 4720
9.584 RuS-PKU-SSPA 4730
9.585 S3Satcom-ACU 4736
9.586 SCC-Device 4742
9.587 SES-DAKS 4745
9.588 SES-Sysinfo 4747
9.589 SNG-Pointing 4752
9.590 SNMP-Driver-Template 4757
9.591 SPX-Etsiline-CommPact-v101 4759
9.592 SPX-Etsiline-CommPact 4761
9.593 SSE-ASat 4765
9.594 SSE-KStar 4769
9.595 SVS-SWC100 4773
9.596 Sampler 4775
9.597 SatService-ACU-IDU 4782
9.598 SatService-ACU-ODM-Serial 4783
9.599 SatService-ACU-ODM 4798
9.600 SatService-ACU-Prediction 4812
9.601 SatService-ACU2-Serial 4818
9.602 SatService-ACU2 4835
9.603 SatService-Beacon-Receiver 4858
9.604 SatService-Deice 4872
9.605 SatService-IO-FEP-Protection-1-To-1 4880
9.606 SatService-IO-FEP-Protection-2-To-1 4883
9.607 SatService-IO-FEP 4885
9.608 SatService-IO-FEP2 4891
9.609 SatService-LBRX-1MT 4901
9.610 SatService-LFRXTX 4904
9.611 SatService-Matrix 4923
9.612 SatService-Power-Sensor 4934
9.613 SatService-SMU 4935
9.614 SatService-Temp-232 4969
9.615 Satellink-1to2-Redundant-LNA 4970
9.616 SatelliteSystems-3430-BTR-L 4972
9.617 Scopus-E1X00-SNMP 4973
9.618 Scopus-IRD26XX 4979
9.619 Scopus-IRD28XX 4985
9.620 Scopus-IRD2900-SNMP 4991
9.621 Scopus-UE9000-SNMP 4998
9.622 Selector-Two-Way 5002
9.623 Selector 5007
9.624 Sencore-AG4400 5011
9.625 Sencore-Afiniti-Encoder 5019
9.626 Sencore-MIP6210 5026
9.627 Sencore-MRD3187-DVBS2 5031
9.628 Sencore-MRD3187-QUAD 5038
9.629 Sencore-MRD5800 5048

9.630 Sencore-MRD7000 5062
9.631 ServerTech-PDU-Sentry-3 5078
9.632 ServerTech-PDU-Sentry-4 5109
9.633 Show-Orbit 5177
9.634 Sicilsat-S00761 5178
9.635 Site-Switch 5182
9.636 Spacepath-CM2-TWTA 5187
9.637 Spacepath-TWTA-Serial 5193
9.638 Spacepath-TWTA 5198
9.639 Spacepath-Tango-TWTA 5204
9.640 Speciality-BDC-Controller 5215
9.641 Speciality-Controller-1926-100 5217
9.642 Speciality-Controller-9753-500 5219
9.643 Spectrum-QAccess 5222
9.644 Store-Float 5225
9.645 Stream-KeySelect 5226
9.646 Switch-Cross-Link 5230
9.647 Symmetric-Site-Switch 5235
9.648 Symmetricom-4036B 5241
9.649 Symmetricom-4059B 5243
9.650 Symmetricom-9611B 5245
9.651 Symmetricom-S650 5247
9.652 Symmetricom-XLI 5250
9.653 Sync-Switch 5254
9.654 Sysinfo 5257
9.655 System 5263
9.656 T21-9261-Decoder 5267
9.657 T21-9261-Encoder 5276
9.658 TLE-Updater 5287
9.659 TX-Channel-ByChannel 5289
9.660 TX-Channel-BySatellite 5300
9.661 TX-Channel 5311
9.662 TX-ChannelSelect-CDB 5328
9.663 TX-ChannelSelect 5334
9.664 TX-LineSettings-CDB 5339
9.665 TX-OFF-Timer 5350
9.666 Tadiran-Scopus-E1000 5352
9.667 Tandberg-Alteia 5356
9.668 Tandberg-E57XX-SNMP 5363
9.669 Tandberg-RX1290-SNMP 5376
9.670 Tandberg-RX8200-SNMP 5383
9.671 Tandberg-RX8200-TriBand-SNMP 5403
9.672 Tandberg-SM62xx 5420
9.673 Tandberg-TT1260-SNMP 5423
9.674 Tandberg-TT1260 5428
9.675 Technotrans-HC-Vertex 5433
9.676 Teledyne-QFlex-400 5442
9.677 Teledyne-QMultiFlex 5449
9.678 Temperature-Converter 5465

9.679 Teracue-TAC 5466
9.680 Terrasat-IBUC-SNMP 5483
9.681 Terrasat-IBUC 5489
9.682 Terrasat-RX-Protection 5493
9.683 Threshold 5495
9.684 TimeLink-Distributor-TMD 5497
9.685 TimeLink-Distributor 5499
9.686 TimeLink-GNSS-Receiver-TMG3400 5500
9.687 TimeLink-GPS-Receiver 5502
9.688 TimeLink-TMG3420 5504
9.689 Timetech-Distributor 5508
9.690 Timetech-GPS-Receiver 5510
9.691 Toggle-Switch 5512
9.692 Transmit-Chain-Preset 5516
9.693 Triple-Translator 5517
9.694 TrippLite-PDU 5519
9.695 Uplink-Power-Control-2 5529
9.696 Uplink-Power-Control 5532
9.697 V4L-DVBS2-Demod 5538
9.698 Vaisala-WXT 5542
9.699 Vertex-7134-Step-Track 5546
9.700 Vertex-7200-Step-Track-DeltaPol 5550
9.701 Vertex-7200-Step-Track-Prec 5556
9.702 Vertex-7200-Step-Track 5562
9.703 Vertex-ACU5000 5569
9.704 Vertex-ACU8100 5577
9.705 Vertex-Digital-Tracking-Receiver 5587
9.706 Vertex-Downconverter-legacy 5590
9.707 Vertex-RSC-1100 5593
9.708 Vertex-RSC-1200 5596
9.709 Vertex-SCR-Downconverter-SNMP-legacy 5598
9.710 Vertex-SCR-Downconverter-SNMP 5601
9.711 Vertex-SCR-Downconverter 5604
9.712 Vertex-SCR-Upconverter-SNMP-legacy 5607
9.713 Vertex-SCR-Upconverter 5610
9.714 Vertex-SSPA 5613
9.715 Vertex-TWTA-2100TK 5617
9.716 Vertex-Upconverter-legacy 5621
9.717 ViaLite-HRA-Dual-EDFA 5624
9.718 ViaLite-HRA-EDFA 5633
9.719 ViaLite-HRS-Switch 5639
9.720 ViaLiteHD-FO 5643
9.721 ViaSat-4100-ACU 5666
9.722 ViaSat-EBEM 5679
9.723 VioIP-End-Point 5685
9.724 Viprinet-HUB-SNMP 5690
9.725 Viprinet-HUB 5692
9.726 Viprinet-Hub-Redundancy-HTTPS 5694
9.727 Viprinet-Router-HTTPS 5696

9.728 Viprinet-Router-SNMP 5702
9.729 Viprinet-Router 5705
9.730 Viprinet-Rugged-Router-HTTPS 5708
9.731 Viprinet-Rugged-Router-SNMP 5715
9.732 Viprinet-Tunnel-SNMP 5719
9.733 Viprinet-Tunnel 5723
9.734 Virtual-Matrix-128x128 5727
9.735 Virtual-Matrix-128x32 5746
9.736 Vitec-MGW-Encoder 5754
9.737 Vocality-FourWire 5764
9.738 WG-Switch 5769
9.739 WISI-Chameleon-DVBC 5773
9.740 Walton-Deice 5777
9.741 Wavestream-BUC-SNMP 5780
9.742 Wavestream-BUC 5783
9.743 Wegener-UNITY44XX 5785
9.744 Wilmers-blueberryCOMPACT 5791
9.745 Work-AT61-Modulator 5793
9.746 Work-Amplifier 5802
9.747 Work-DualSBD 5805
9.748 Work-DualSBU 5808
9.749 Work-DualSSBD 5811
9.750 Work-DualSSBU 5814
9.751 Work-DualVSCD 5817
9.752 Work-DualVSCU 5821
9.753 Work-MBU-MiniBuc 5825
9.754 Work-QuadSBD 5827
9.755 Work-QuadSBU 5830
9.756 Work-RSC 5834
9.757 Work-RSCM-Dual 5835
9.758 Work-RSCM-Quad 5843
9.759 Work-RSCM-Triple 5853
9.760 Work-RSCM 5862
9.761 Work-SBD 5869
9.762 Work-SBU 5872
9.763 Work-SCD 5875
9.764 Work-SCU 5878
9.765 Work-SDM2-MT6-Modulator 5881
9.766 Work-SDM2-Modulator 5899
9.767 Work-SM2CU-Modulator 5904
9.768 Work-SSBD 5909
9.769 Work-SSBU 5911
9.770 Work-STLT2-KuK 5913
9.771 Work-TLT-LO 5916
9.772 Work-TLT 5919
9.773 Work-TriSBD 5923
9.774 Work-TriSBU 5926
9.775 Work-TripleVSCD 5929
9.776 Work-VCR 5933

9.777 Work-VSCD 5935
9.778 Work-VSCSR 5938
9.779 Work-VSCU 5940
9.780 Work-VSM2CU-Modulator 5944
9.781 Work-VSSBD 5969
9.782 Work-VSSBU 5972
9.783 XMW-Blockupconverter 5974
9.784 Xicom-KHPA 5976
9.785 Xicom-TWTA-XT 5981
9.786 Xicom-TWTA-XTD 5983
9.787 Xicom-TWTA-XTDL 5989
9.788 Xicom-TWTA-XTRT 5996
9.789 Xicom-TWTA 6002
9.790 Xytronix-X600 6007
9.791 Zodiac-Cortex-ACU 6010
9.792 Zodiac-Cortex-CRT-IFM 6013
9.793 Zodiac-Cortex-CRT-IFR 6021
9.794 Zodiac-Cortex-CRT-PRJ 6027
9.795 Zodiac-Cortex-CRT-TCU 6031
9.796 Zodiac-Cortex-CRT-TMS 6037
9.797 Zodiac-Cortex-CRT-TMU 6042
9.798 Zodiac-Cortex-CRT 6050
9.799 Zodiac-Cortex-DTR-Cal 6062
9.800 Zodiac-Cortex-DTR-Path 6064
9.801 Zodiac-Cortex-DTR 6066
9.802 Zodiac-Cortex-Reset 6070

This is the ***sat-nms*** MNC Monitoring & Control System user manual, describing the operation, configuration and installation of all components of the ***sat-nms*** MNC software framework.

This version of the user manual is the complete edition with all topics included. For the convenience of the reader we also provide splitted manuals covering on a subset of ***sat-nms*** MNC system.

These subsets are:

- ***sat-nms*** MNC WebClient user manual
- ***sat-nms*** MNC Legacy Java client user manual
- ***sat-nms*** MNC Installation and administration manual
- ***sat-nms*** MNC API documentation
- ***sat-nms*** MNC Universal device driver language reference manual
- ***sat-nms*** MNC Device driver reference

Support and Assistance

If you need any assistance regarding the ***sat-nms*** software, do not hesitate to contact us. We would be pleased to help you.

satnms-support@satservicegmbh.de

SatService GmbH
Hardstrasse 9
78256 Steisslingen
Germany
phone +49 7738 99791-10
www.satnms.com

Version 3.80.6-a -- 2025-12-03 -- © 2003-2025 SatService GmbH

1 Overview

Abstract

This is the online user manual describing the ***sat-nms MNC*** monitoring & control system.

There is also a PDF version of this document, but we recommend to read the documentation online, as the user manual makes extensive usage of hyper-links which help you to jump quickly to topics which are related to that one you're actually reading.

The main sections of the user manual are:

- M&C Operation with Web Client The easy to use web browser based user interface the The ***sat-nms*** M&C System contains manifold features and functions. This section of the manual first gives a short introduction, describes the user interface and functions in detail.
- M&C Operation with Legacy Java Client Beside the new web client the system supports the legacy Java client application. This chapter describes the functionality of this client for your reference in existing installation not upgrade to the ***sat-nms*** web client.
- Administration This chapter explains in details aspects of the administration of all components of the ***sat-nms*** Monitoring & Control System. This includes for example backups, updates and a reference for all configuration files.
- Installation Usually SatService delivers pre-installed system with all components ready to use. This chapter is indended for
  the case that software components should be installed on customer provided bare metal servers or virtual machines and to provide a better understanding about the different components.
- Device Driver Reference Device drivers are the bridge between equipment and M&C system. The ***sat-nms* Universal Device Driver Language described in this chapter allow to define these drivers.
- API Reference the ***sat-nms*** backend service provides a REST-API to all function and data of the M&C system. This chapter is the complete reference to all API end points and data structures.
- Device Driver Descriptions This chapter include a full list of all device drivers with a short help page how to setup these devices and which parameters are available. Beside this it contains for logical devices detailed instructions on how to use them.

If you are not familiar with this Web based online help system, you may want to read the following chapter How To Use The Online Help, which briefly explains how to navigate thru the online help.

1.1 Using The Online Help

The user manual of the ***sat-nms*** software is built as a large set of hyper-linked web pages. These help pages are available

- on installed M&C systems via web and java client

- in the Internet at <www.satnms.com> or <www.satservicegmbh.de>
- inside the installation package `satnms4-documentation`

Invoking The Online Help

Most pages in the web client and java client contain a help button in This button opens the web browser installed on your computer and shows the help page explaining the features of the window where you clicked to the help button. Once you are in the online help system, you may use the hyper-links to navigate.

The Navigation Buttons

Every documentation page show three navigation buttons right beside the text which let you move thru the manual:

Function	Icon	Description
next		Advances to the next page within the logical structure of the user manual. This is useful if you want to read the online help like a book, page by page.
back		Goes one page back. Don't confuse with the browsers `back` button, which recalls the page you recently have watched.
menu		Jump to the table of contents page.

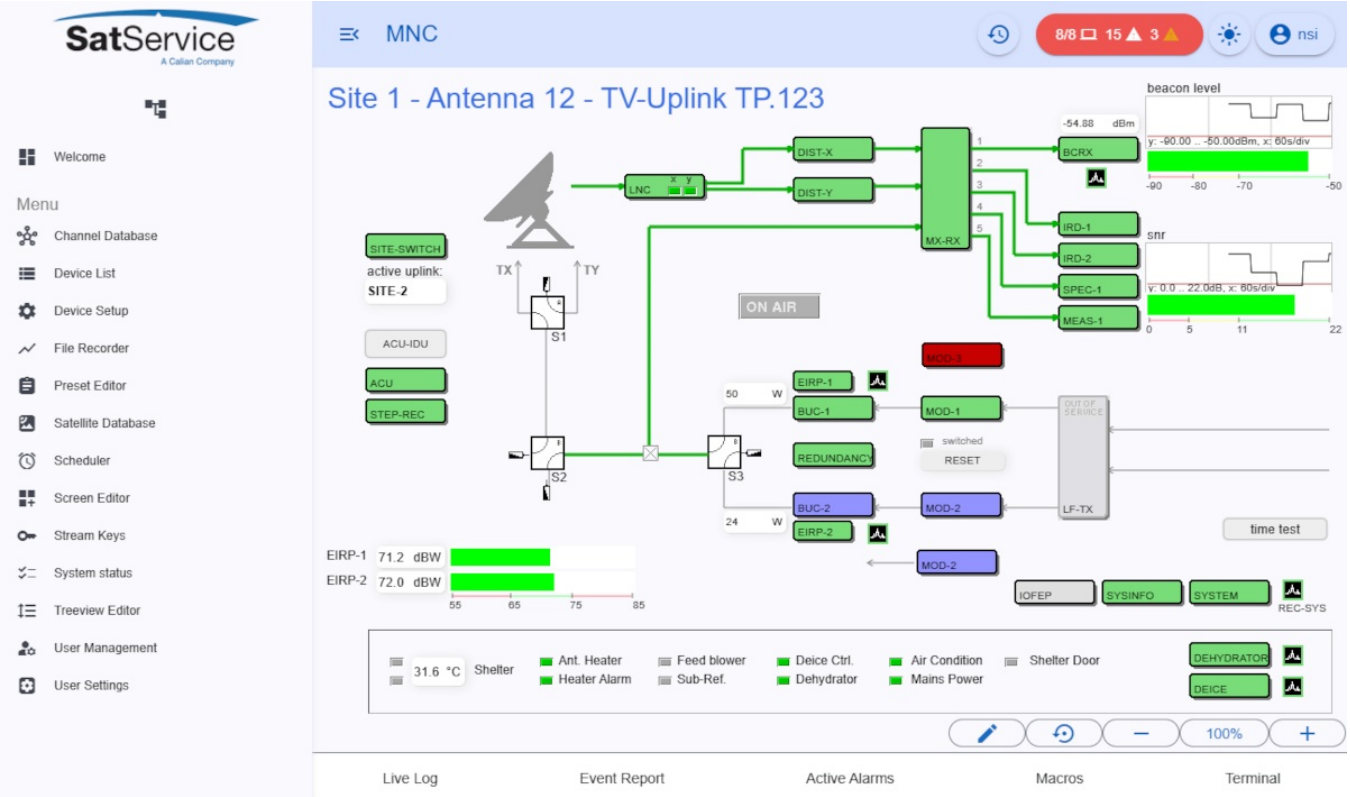
1.2 Architecture

The following chapters will give an overview about the **sat-nms** MNC Monitoring & Control system architecture and all the components.

TODO

2 M&C web-client

This section describes the functionality and the usage of the sat-nms M&C software when operated with the web-client. The following chapters of the manual describe each function and each user interface screen in detail.

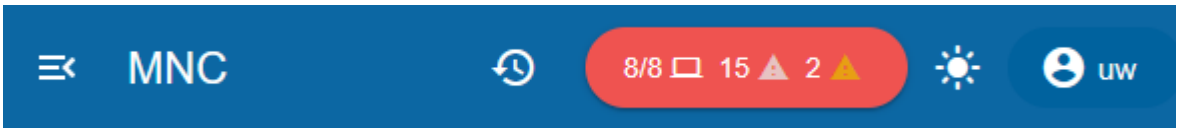


Quick access links to the main sections of the document:

- Top toolbar and Sidebar / Main Menu ok!!!
- Main window ok!!!
  - Display elements ok!!!
  - Device details panel ok!!!
  - Device window ok!!!
- Treeview ok!!!
- Channel database ok!!!
- Device list ok!!!
- Device setup
- File recorder
- Preset editor
- Satellite
  - Satellite list
  - Satellite operator list
  - Antenna
- Macro scheduler

- Screen editor
- Stream keys
- System status
- Treeview editor
- User management
- User settings
- Bottom panel

2.1 Top toolbar



Located on top of the page and permanently visible and it provides quick access to the user menu, MNC notifications etc. It can change the color from blue to red if you have configured that in user-settings warning section to show the fault state of the system.

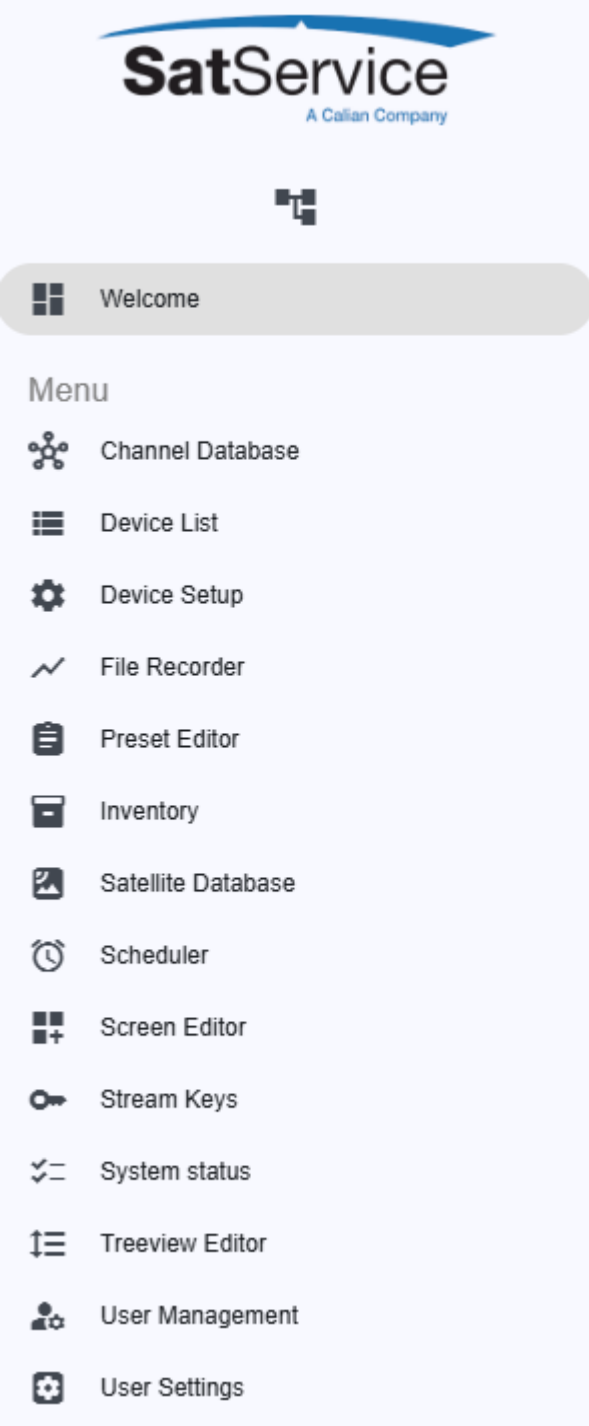
The following functions are available by clicking on the elements:

icon	description
	toggles visibility of sidebar (main menu) on the left
	shows the page history
notification area	show list of MNCs and active faults
	switch UI theme (dark mode)
	opens the user menu

2.1.1 Sidebar / main menu

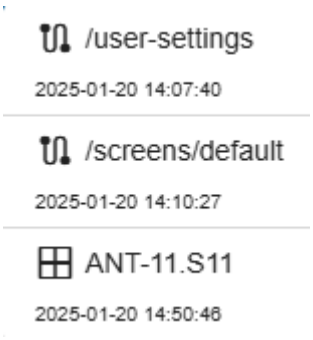
The side bar provide shortcuts to the different functions of the***sat-nms*** MNC system. It a kind of menu menu which gives you quick access to key areas and can be find on the left hand side. All menus are explained in the separates chapters.

You can hide the sidebar by clicking on the button in top toolbar



2.1.2 History

Clicking on the history button opens the list with your navigation history. If you click on a list entry you jump directly back to the listed page. This could be page from the main menu, a user screen or a device window.



Each list entry show the type (by icon), the name of the page (URL) or the name of the device. The timestamp in UTC corresponds to the time when the item was visited.

icon	description
	jump to a page or user screen (URL)
	opens specified device window

Remark: After reloading the page, the history is deleted.

2.1.3 Notification area

It gives you a clear view to the state of all connected servers and devices. Starting from left: The background color shows the summary fault state of the complete installation. The color is priorized in the following order (from low to high):

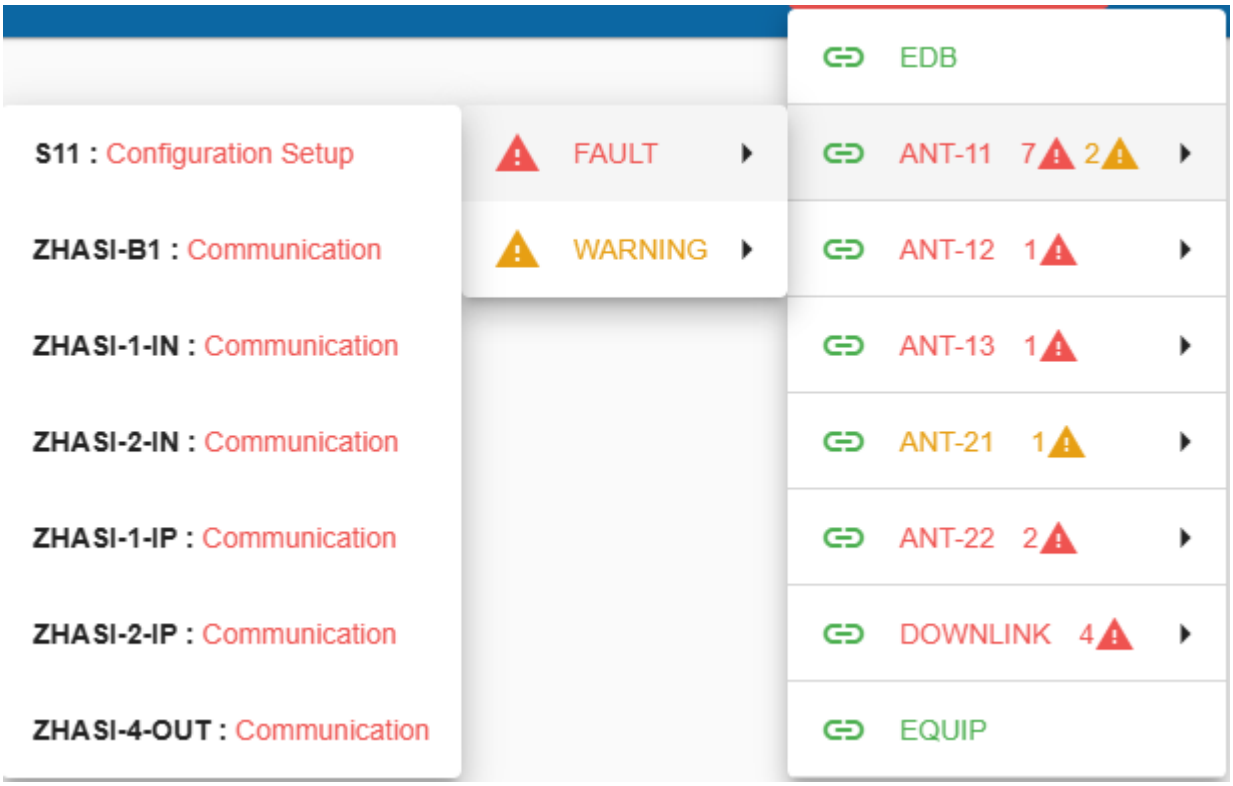
- **green**: no active alarm or warning
- **orange**: at least one device is in warning state
- **red**: at least one device is in fault/alarm state

The number inside the notification represents the following information:

- **MNC count**: number of MNC servers is displayed on the left-hand side of the computer icon
  - left number: currently connected/reachable MNCs
  - right number: total number of MNCs
  - e.g. 6/10 means 6 of 10 MNC services are currently reachable
- **Alarm count**: total number of active alarms
- **Faults**: total number of active faults
- **Warnings**: total number of active warning

If there is a new fault arriving, the corresponding symbol will be start blinking. Clicking on it will stop blinking and/or stop alarm tone.

On click it shows a dropdown list with MNC names with details about MNCs and active faults



The first level of this drop down menu show the list of configured MNC services. Clicking on such an entry will open the default user screen associated with this MNC (if configured)

- **Status indication (Icons)**
  - **Green link icon**: The MNC is connected.
  - **Red link off icon**: The MNC is disconnected.
- **Green name**: MNC have no alarms, faults and warnings.
- **Red name**: MNC have alarm(s) or fault(s).
- **Yellow name**: MNC have warning(s).
- Numbers beside each MNC name indicate the total of alarms (red), faults (red) and warnings (yellow).

When a MNC has alarms, faults or warnings, a secondary dropdown appears with the following options:

- **Alarms/Fault/Warning** per MNC
  - **ALARMS**: Displays a list of devices with alarms.
  - **FAULT**: Displays a list of devices with faults.
  - **WARNING**: Displays a list of devices with warnings.

Selecting the error type opens the 3rd level with a list of affected devices. Clicking on an entry will open the matching device in the device details panel.

After selecting the type `ALARMS`, `FAULT` or `WARNING` a list of affected devices is shown along with a brief error description

2.1.4 Theme

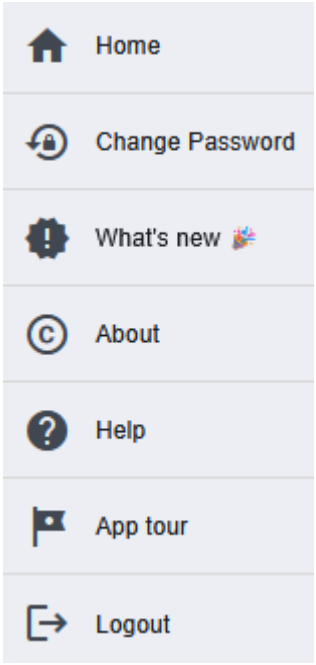
This icon shows current theme of this app. On clicking shows the following options:

- **Auto**: App will adapt the system theme.
- **Dark**: Activates dark mode.
- **Light**: Activates light mode.

2.1.5 User menu

Near the profile icon is the logged-in username. By clicking on name the user menu expands and provides the following options:

- **Home**: navigate to the default page.
- **Change password**: open the dialog to change the password.
- **What's new**: new features and improvements are listed in this sections. It's open a panel on right hand side with details.
- **About**: open about page. It contains app information like company name, address, app version etc.
- **App tour**: start the app tour and navigates through the app and it's features.
- **Logout**: logout the current user.



2.2 Main Content Window

The Main Window is the central area of the application where all content and pages are displayed. It serves as the primary **workspace** where users interact with the application’s functionality. Whether it's data views, editors, user-screens, or any other content, everything is presented within this section.

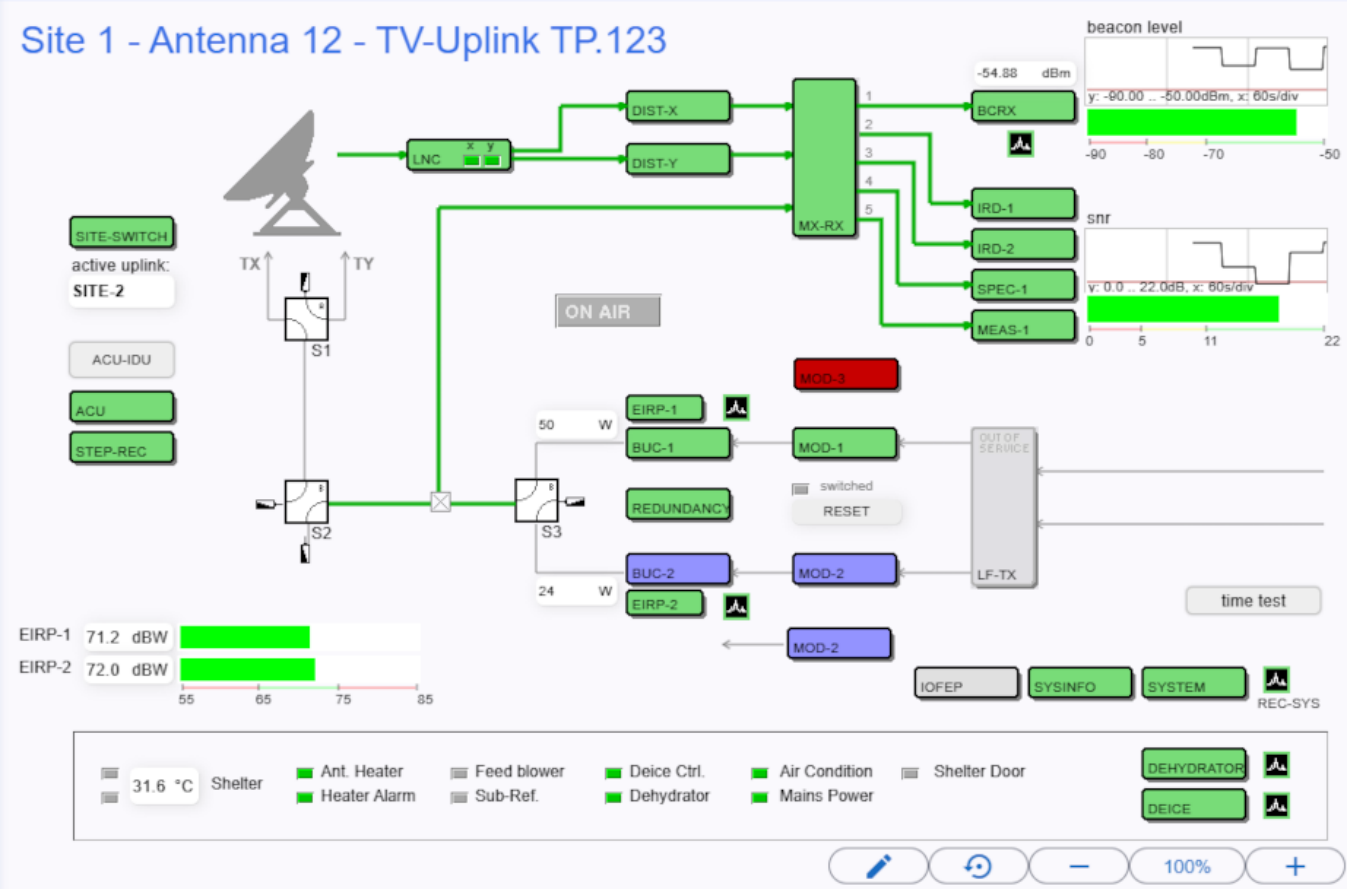
- The Main Window adapts dynamically based on the selected page or action.
- It works together with the top toolbar, the sidebar and the bottom toolbar to ensure a seamless user experience.
- Users can navigate between different sections using the sidebar menu or other navigation controls, and the content in the Main Window updates accordingly.

Following content types are shown in the Main Window:

- sidebar menus functions (expect tree view)
- user screens
- device windows
- all dialogs

2.2.1 User screens

The User Screens are the screens that are configured withthe screen editor. You can interact with the user screen (zoom in or out, scroll vertically or horizontally, reposition the entire screen) and the elements (depending on the element). You do not have to reload page to get the actual values. The values of the elements are always updated. If you move the mouse over an element, details about this element are displayed on left bottom corner of main window. However, you must activate this setting in the user settings under 'Debug mode'.



In the bottom right-hand corner of themain window there is a toolbar with the following

functions:

Function	Icon	Description
edit screen		open the screen editor with the active user screen
Reset position		resets the position of the user screen
Zoom out		zoom out, scale down the elements inside of the page
Current zoom		display the current zoom of the user screen. Clicking on it works like a button and resets the zoom to the default setting.
Zoom in		zoom in, scale up the elements inside of the page

User screens shortcuts

- **Zooming**
  - ctrl + + to zoom in.
  - ctrl + - to zoom out.
  - ctrl + 0 to rest zoom.
  - ctrl + mouse wheel to zoom in and out.
- **Scrolling**
  - Mouse wheel for vertical scroll.
  - shift + mouse wheel for horizontal scroll.
- **Drag and drop**
  - Space + hold mouse left click + mouse move to drag and drop the user screen on desired position.

2.2.2 Display elements

This section described briefly all the display elements e.g. interaction, state behavior, disable type etc. If you want to check the display element properties please check the element properties in Screen Editor chapter.

Category	Element	Readonly	Interaction	Dynamic update	Update data
Static	Text	□	□	□	□
	Frame	□	□	□	□
Dynamic non-interactive	Display (Readonly)	□	□	□	□
	Icon	□	□	□	□

Category	Element	Readonly	Interaction	Dynamic update	Update data
	Arrow	□	□	□	□
	Rect	□	□	□	□
	Gauge	□	□	□	□
	Thumbnail	□	□	□	□
Interactive	Parameter	□	□	□	□
	Parameter button	□	□	□	□
	Latching button	□	□	□	□
	Radio button	□	□	□	□
	Button	□	□	□	□
	Device icon	□	□	□	□
	Switch icon	□	□	□	□
	Chart	□	□	□	□
	XY chart	□	□	□	□
	AzEl	□	□	□	□
	Spectrum	□	□	□	□
	Target list	□	□	□	□

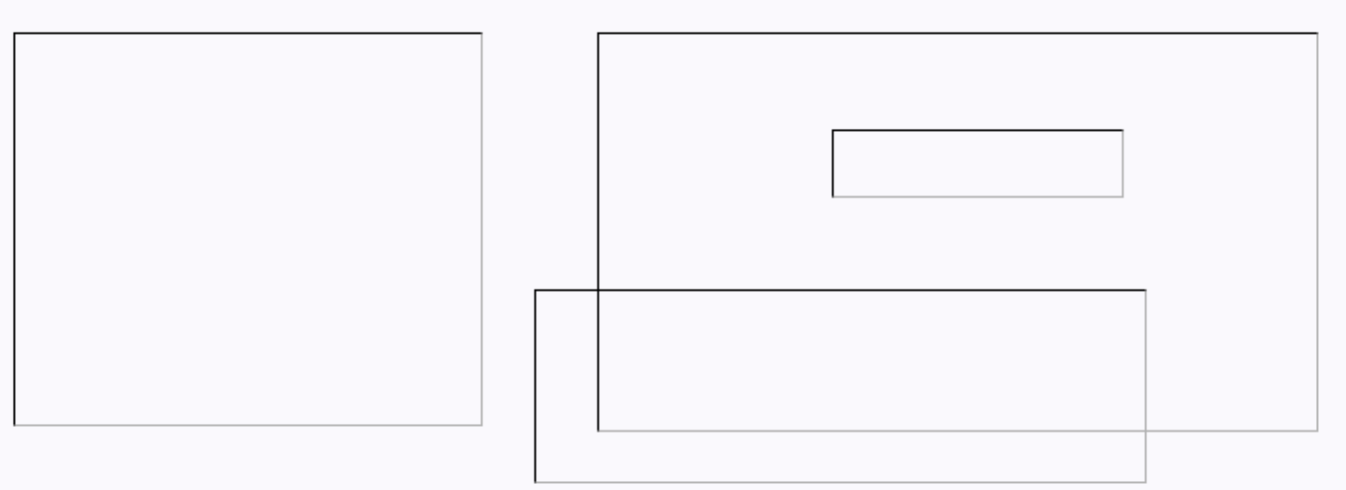
- □ : Yes
- □: No
- □: Element can be restricted with the privilege level.

2.2.2.1 Text element

The text (label) element is an static type of element and can not change their values until you changed it from Screen-Editor. It can have different color, font-style and font-size.

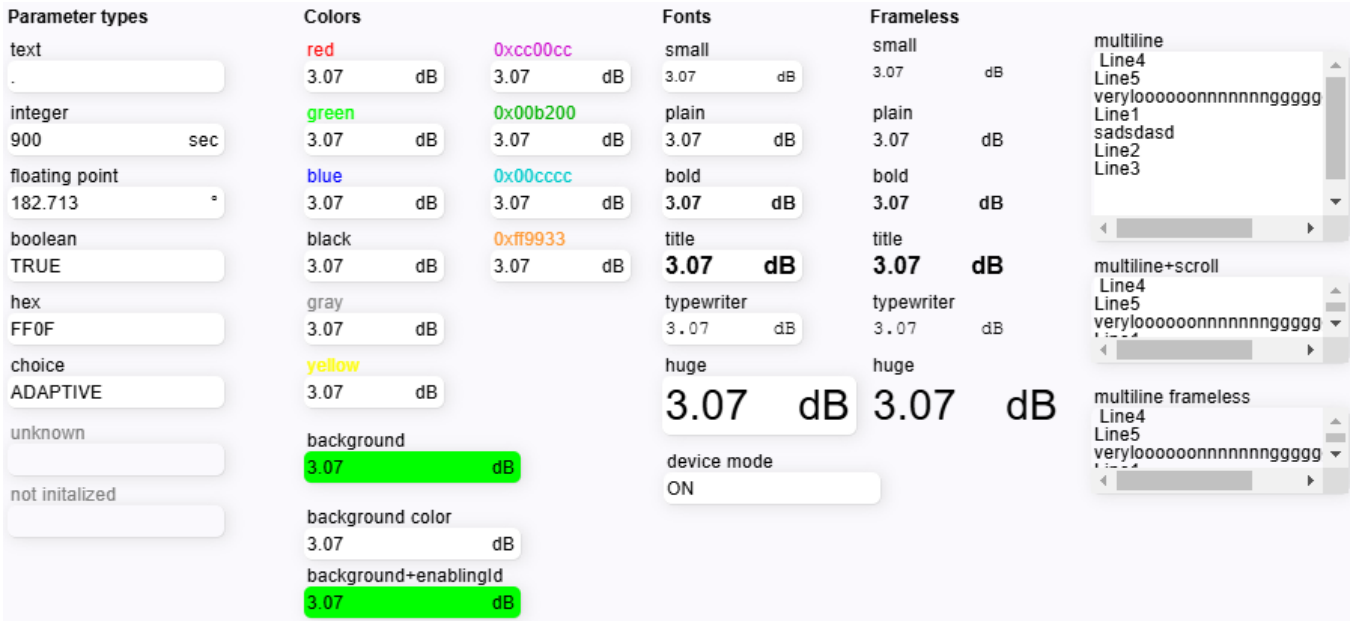
2.2.2.2 Frame element

The Frame element is an static type of element and draws a sunken 3D frame, which is intended to be used to group other elements.



2.2.2.3 Display element

This display element also called read-only element is an dynamic non-interactive element. That means you can not interact with the element to edit or send values. But it still get's updated value. This element is most like parameter element only with the difference that you can not interact with it.

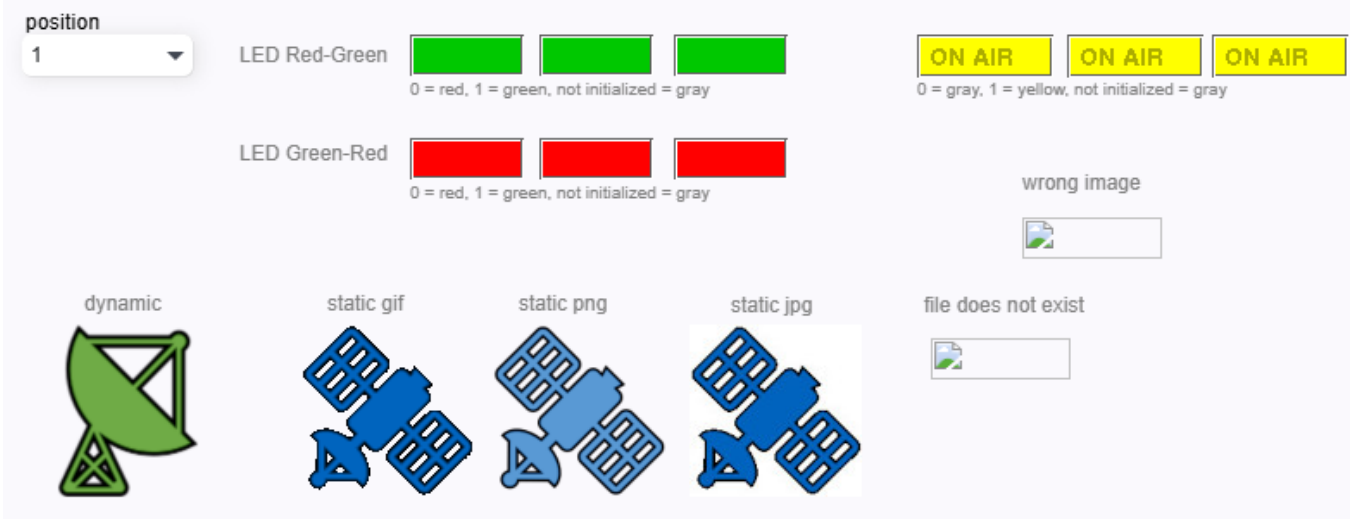


The image above shows the different types of color, fonts, types, background etc. to get an overview of possible looks of the display element (readonly element). Additional colors or background colors can be configured in the element properties in the screen editor.

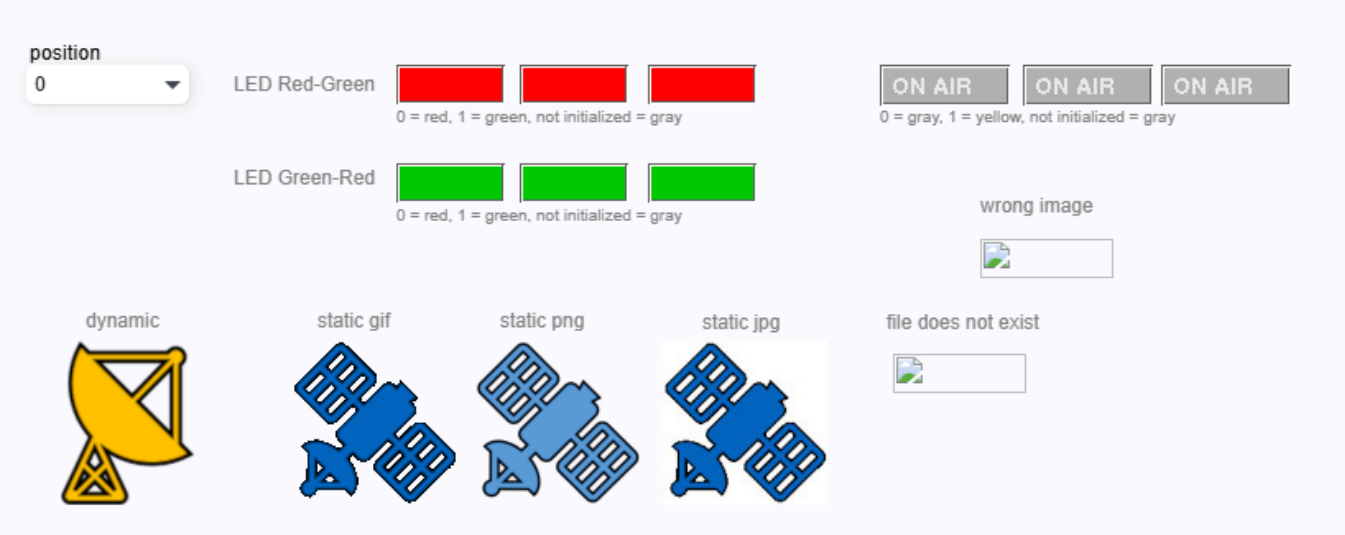
2.2.2.4 Icon element

The icon element works as a dynamic non-interactive element and also as a static element. If message ID in the element properties configured, it will works like an static element type, otherwise it works like dynamic non-interactive element.

The images below shows static and dynamic icons.



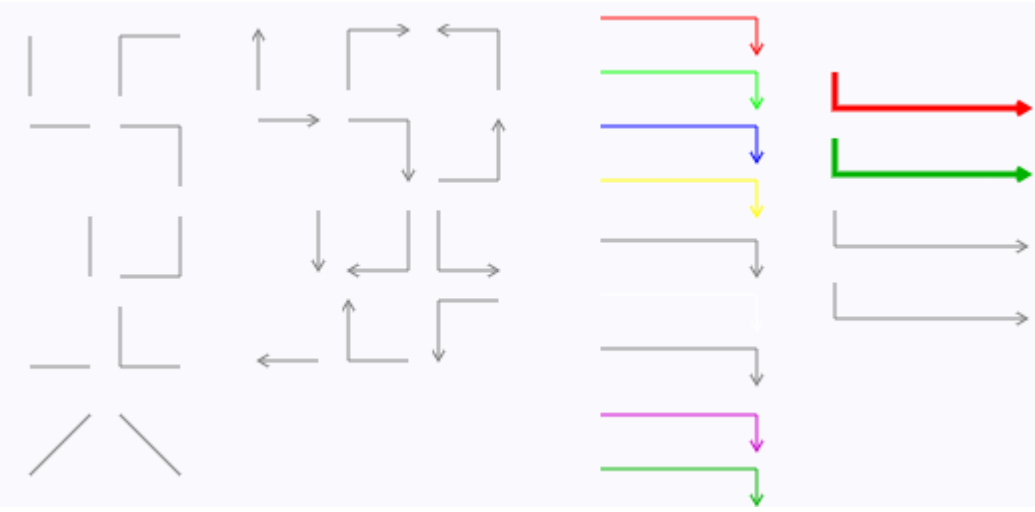
Dynamic icons are bind with position parameter element.



NOTE: Please be careful on selecting an icon with message ID. Wrong image heading in above image is better example of this. Some of the icons contain the name of 0, 1, on or off. For example in above page parameter value can be between 0 and 1. The icon will be changed to parameter value.

2.2.2.5 Arrow element

The Arrow element work as a dynamic non-interactive element and also as a static element. It draws a horizontal and/or a vertical line and optionally an arrowhead. The arrow/line's color is selectable, also may follow a parameter value. You can configure the element properties in the screen-editor.



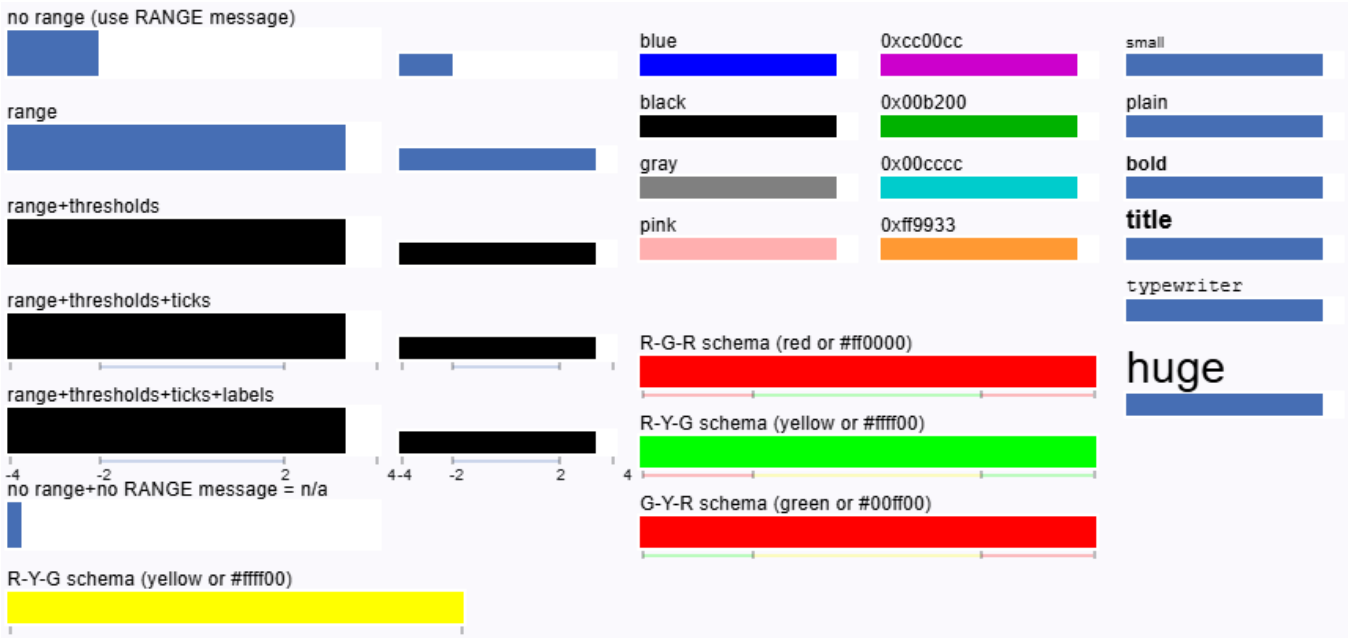
2.2.2.6 Rect element

The Arrow element work as a dynamic non-interactive element and also as a static element. It draws a rectangle with a selectable color. The element may listen to a parameter value and change the color of the rectangle according to the parameter value.

2.2.2.7 Gauge element

The gauge element is an non interactive element and shows a numeric parameter value as a horizontal bar in an entry field like frame. The gauge element is capable to adjust the scale factor for the gauge automatically from the parameter's range definition. Alternatively the scale parameters may set explicitly.

Image below shows the gauge elements with different type of element properties.

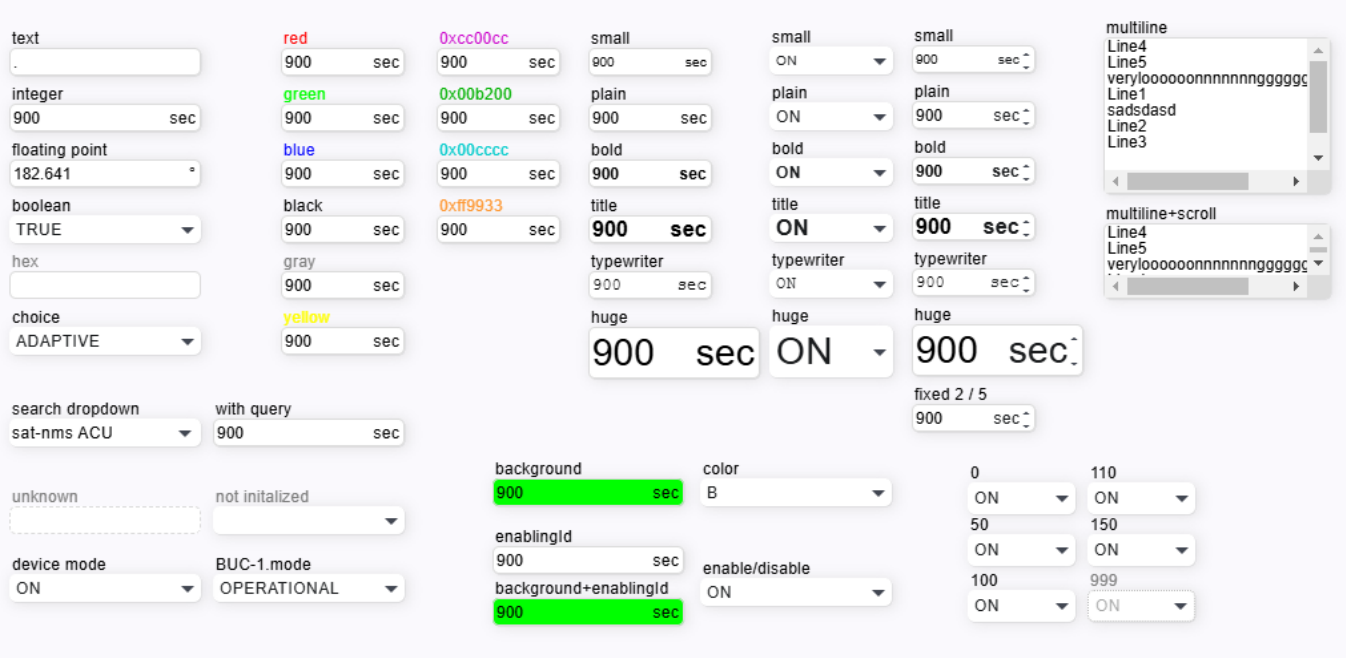


2.2.2.8 Thumbnail element

2.2.2.9 Parameter element

The ParameterElement is the common component to display and edit most types of M&C parameters. Depending on the data type of the parameter (the data type is detected automatically) the parameter elements appears as textual / numeric entry field, choice box or as display field for read only parameters.

In the image below are the all variants of parameter element.



Widget Variants: Depending on the type of parameter and on the*height* parameter there are different widget types associated with the ParameterElement. As the parameter type is not known at the time the screen definition is read, the decision what widget type shall be used must be done after the parameter type description has been received from the server.

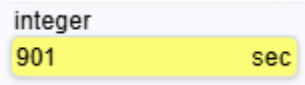
Parameter Type	***height < 40***	***height >= 40***
Text	single line entry field	multi line entry field, scrollable if text does not fit into the field
Numeric	single line entry field	single line entry field
Enum/Choice	drop down or combo box	scrollable list selection
r/o Text	single line display	multi line display, scrollable if text does not fit into the field
r/o Numeric	single line display	single line display
r/o Enum/Choice	single line display	single line display

Spin Buttons: With *useSpinButtons* set 'true', the ParameterElement shows spin buttons with editable numeric values. The spin buttons allow to increment/decrement the value by clicking on them. By default, a spin button click increments or decrements the lowest significant digit

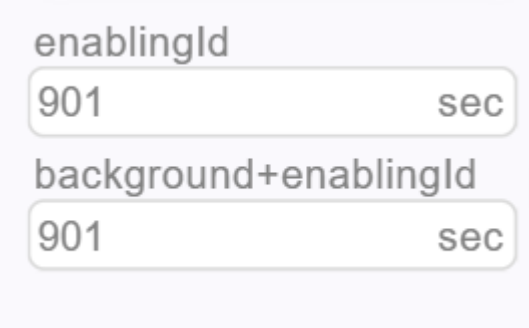
shown in the entry field. With the shift key hold down, the effective increment is x10. The above applies if the *spinSmallIncrement* / *spinLargeIncrement* properties are empty. If set, these values override the defaults.

Unsaved changes

If you have changed the value of any parameter element, the background of the input field turns yellow. As soon as you have saved the changes by pressing the Enter key or leaving the input field, the background color of the input field returns to normal.



Element disabled

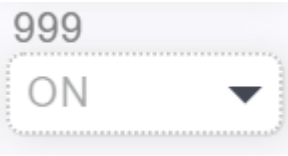


Element readonly

Readonly elements is displayed with dashed border with grey text color.

Element disabled because of low privilege level

Elements is displayed with dotted border with grey text color.



2.2.2.10 Parameter button element

The parameter button element is an interactive element and a button which sends a parameter value if pressed. A frequently used application for the parameter button is a RF-OFF button which sends a "tx.on=OFF" to a certain device. Beside this, a parameter button also may be programmed to play a parameter setting macro.

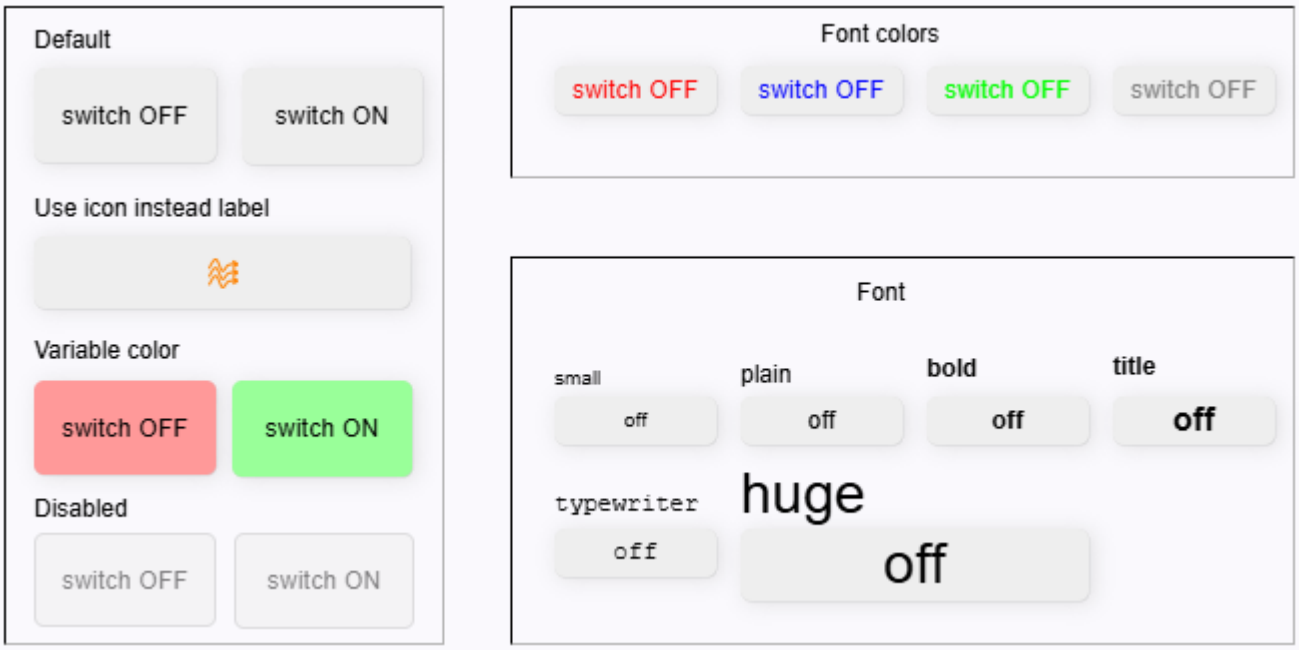
The parameter button element never uses a button label and an icon at the same time. Either

icon or label is an empty string.

If no parameter value is set, the parameter button element sends the button label or the icon name as the parameter value when clicked.

Macro Buttons

The MacroButton is a special version of the parameter button element. A*isMacroButton = true* marks a parameter button element to be a MacroButton. In this case the 'id' field contains the name of the M&C where to play the macro and the macro name in this case, separated by a period.



2.2.2.11 Latching button element

The latching button element works much like the[parameter button element](#) described earlier in this document, but is specialized to show and control an enumeration parameter which knows exactly two states (e.g. on/off or true/false).

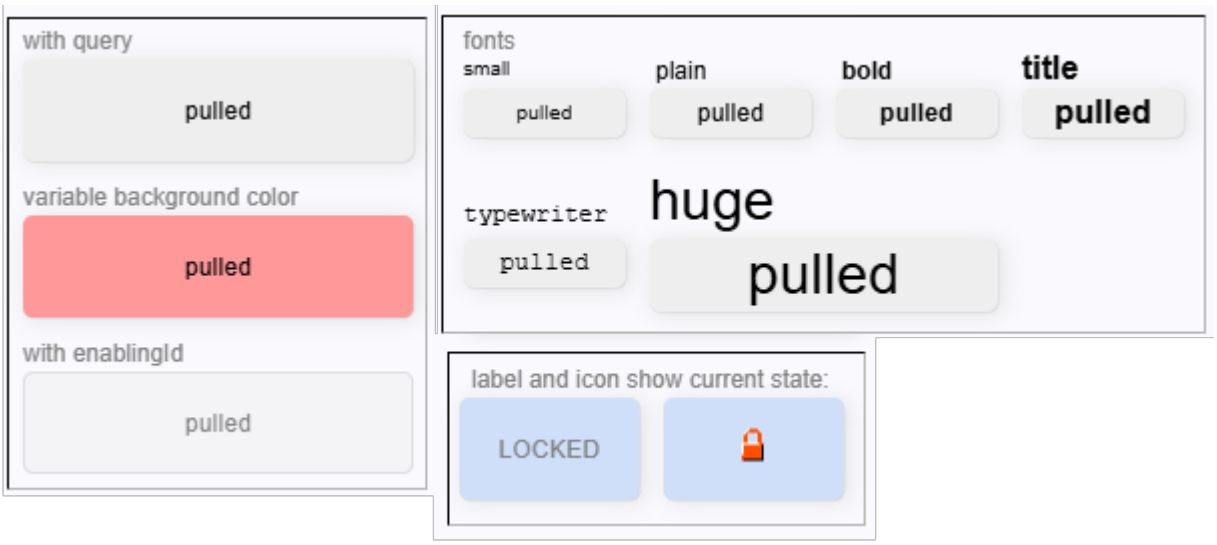
When the operator changes the state of the button by clicking it once, the latching button sends the 'other' parameter value to the device. On the other hand, if some other instance in the system changes the parameter state, the latching button recognizes this and changes the up/down state of the button accordingly.

Like the parameter button, the latching button may be labeled with text or an image. To reflect the actual state, the latching button always is configured with two text string or two image names which are shown according to the actual parameter value.

The latching button element never uses a button label and an icon at the same time. Either iconUp / iconDown or labelUp / labelDown contain empty strings.

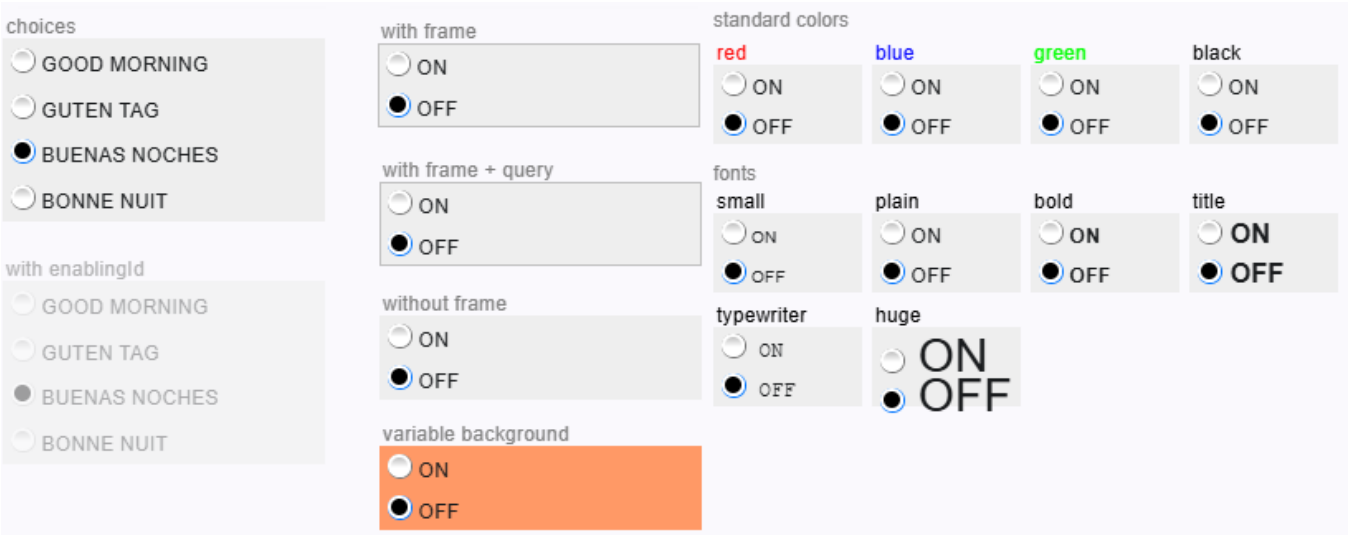
The valueUp and valueDown values are not only sent when the button is pressed or released.

They also are used to set the button state when a parameter value is received though id: If the value matches valueDown the button goes to 'pressed' state, in all other cases the button gets released. Labels or icons change an this moment according to the new state.



2.2.2.12 Radio button element

The radio button element is a component to display and edit CHOICE type M&C parameters as a number of radio buttons. Depending on the height of the element, radio buttons are positioned in a row or in a column (height<=40 means position horizontally). The radio buttons are labeled with the choices of the parameter, there are as many radio buttons as choices in the parameter range.



2.2.2.13 Button element

The button element creates a button which launches another user interface screen if pressed. The screen to be launched may be another user defined screen or a predefined one.

The *action* field of the ButtonElement defines what the button will do when clicked by the operator. The list below shows all known values for *action*, the meaning of this and a description of the parameters *par1* and *par2* which will have different meanings for particular action.

CHILD-SCREEN

Launches a new browser tab or window showing the user screen referenced with*par1*.

- *par1:* The name of the user screen to show.
- *par2:* null
- *par3:* null

REPLACE-SCREEN

Replaces the actual user screen by the one referenced with*par1*.

- *par1:* The name of the user screen to show.
- *par2:* null
- *par3:* null

LOAD-PRESET

Launches a "Load Preset" dialog which lets the operator select and apply a device preset from a list of presets which are defined for this particular type of device. Device presets are local to each M&C instance, have to be loaded and applied from there.

- *par1:* The name of the device to which the preset shall be applied to. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYDEVICE` (must be set in the screen editor this way).
- *par2:* The name of the driver of this device.
- *par3:* An option search pattern to filter the list of displayed presets. All presets containing the given string (not case sensitive compare) shall be shown. If *par3* is null, all preset for the given device / driver combination shall be shown.

FREC-VIEW

Launches a File-Recorder view window. This window shows the data recorded by the File-Recorder device referenced with *par1* or the live data provided by this device.

- *par1:* The name of the File-Recorder device. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MY-FREC` (must be set in the screen editor this way). In the device screen of the file recorder, the special device name "." denotes *this* device.
- *par2:* The file recorder view preset number (0..7) to be applied. May be null, in this case no stored settings are applied.
- *par3:* null

SPECTR-VIEW

Launches a Spectrum Display window. This window in fact is a device screen of the CSM-Spectrum-Analyzer device.

- *par1:* The name of the spectrum analyzer device. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MY-FREC` (must be set in the screen editor this way).
- *par2:* A comma separated list of message-id/value pairs. Within each pair the message-id is separated from the value by one space character. Message ids are fully qualified, starting with the M&C name (must be set in the screen editor this way). The Spectrum Display window shall parse this list and send every message defined in the list with a poke call to the backend. This macro-like function is used to initialize the spectrum analyzer or to switch its input when the window is launched.
- *par3:* null

BROWSER-VIEW

Opens a new Browser window and displays a given URL in this window. This function is used to invoke the sat-nms online help and to launch the Web-GUI of certain devices.

- *par1:* The URL to show. Before launching the browser window, some replacements have to be done on the URL string:
  - If *par1* is a string consisting only of 4 decimal digits, this is a sat-nms online help topic number and the string must be expanded to the full URL from where this topic can be loaded by the browser.
  - If the ButtonElement is part of a device screen and the URL contains '@' characters, they have to be replaced with the (IP-) address of the device. A double '@@' escapes this behavior and gets replaced by a single '@' in the URL string.
- *par2:* null
- *par3:* null

TREE-NAVIGATE

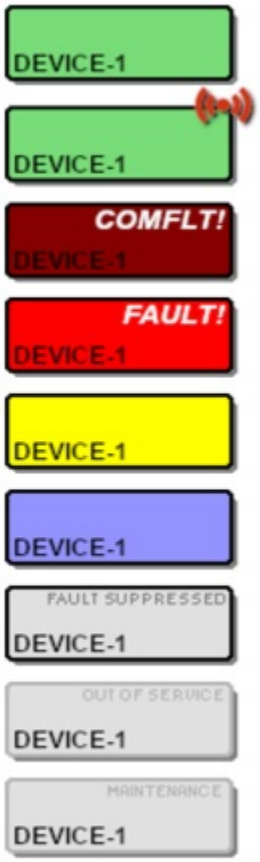
In a tree view UI this button navigates in the tree to the tree path referenced with *par1*. As the tree view UI with the WebUI is still t.b.d., this description should be considered as preliminary.

- *par1:* The tree view path to navigate to.
- *par2:* null
- *par3:* null

2.2.2.14 Device element

The device icon display element represents a device in the M&C user interface. It displays the status of the device by it's color/shape and gives access to the device details panel panel for this particular device by a mouse click.

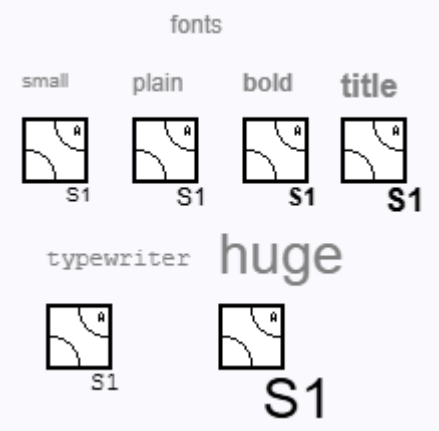
- Operational – ok
  - Device polling active, no alarm/fault
- Operational – ok, transmitting
  - Operational + ok and TX on
- Operational – CommFlt
  - No response from equipment
- Operational – fault
  - Device polling active, alarm/fault active
- Operational – warning
  - Device polling active, warning active
- Redundant unit
  - Operational + ok, Standby unit in a redundancy setup
- Fault-Suppressed
  - Device polling active, faults will be ignored
- Out-Of-Service
  - No polling, no controlling possible
- Maintenance
  - No polling, no controlling possible



(User defined device icons set with different color schemes are available)

2.2.2.15 Switch element

The switch element is a special version of the device element which may be used to visualize the position of a switch in a user interface screen designed as a block diagram. The switch icon has all capabilities of a plain device icon display element. The device details panel opens an you can toggle the position of switch.



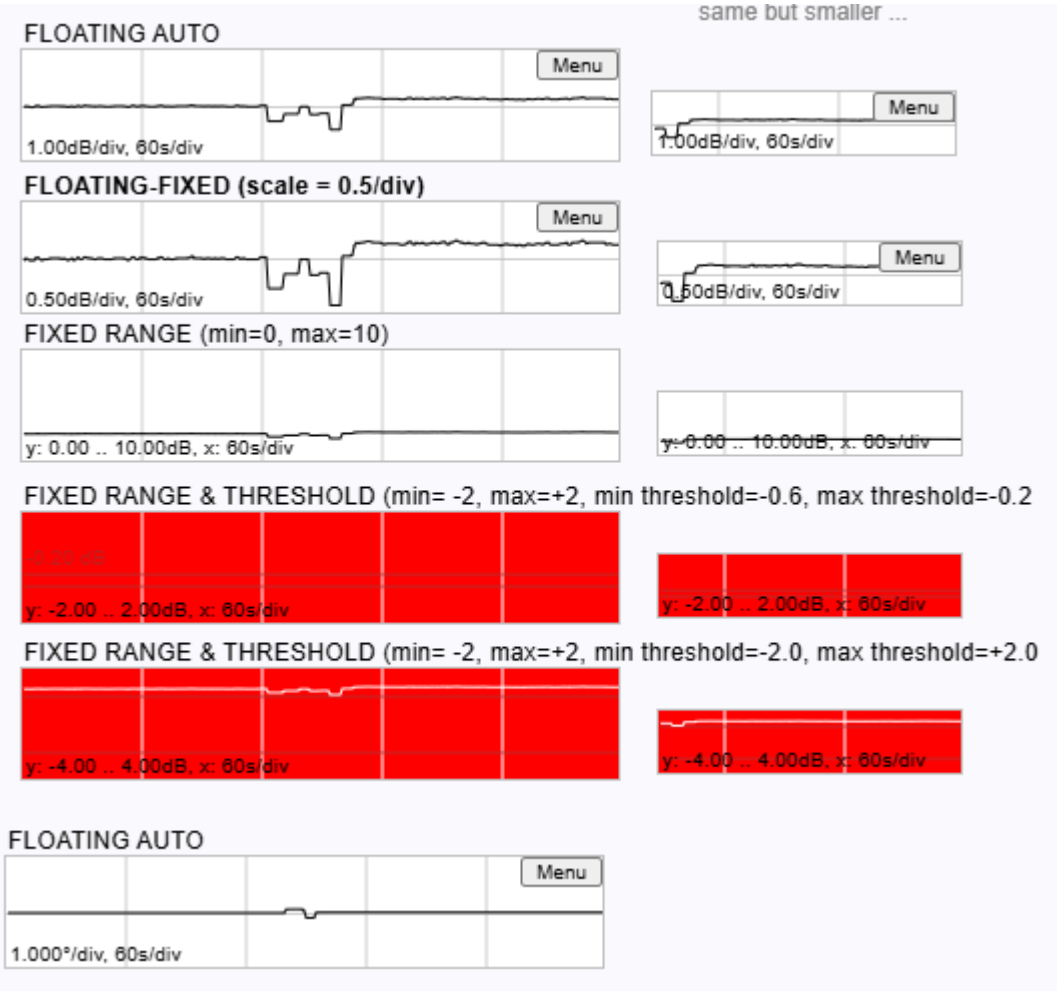
Icon Selection

The switch element displays one icon of a given set depending on the switch position and the state of the switch device. For this the SwitchElement appends a '-' character, a 2-character status code and the suffix '.png' to the icon name given in *icon* in order to ge the file name to load / display:

2.2.2.16 Chart element

The chart element shows a strip chart of a numeric parameter. The chart element keeps a local history of the received values, advances with a constant speed of 1 pixel / second. The default y-scale is 1/division but may be changed by clicking to the chart with the right mouse button.

By default, the strip chart element lets the y-scale offset follow the displayed value that the recent measurement samples are shown in the diagram. This behavior is optimized for applications where the strip chart shall indicate a 'trend' for the displayed value, using an element height of only 50 pixels or less.



Display Modes

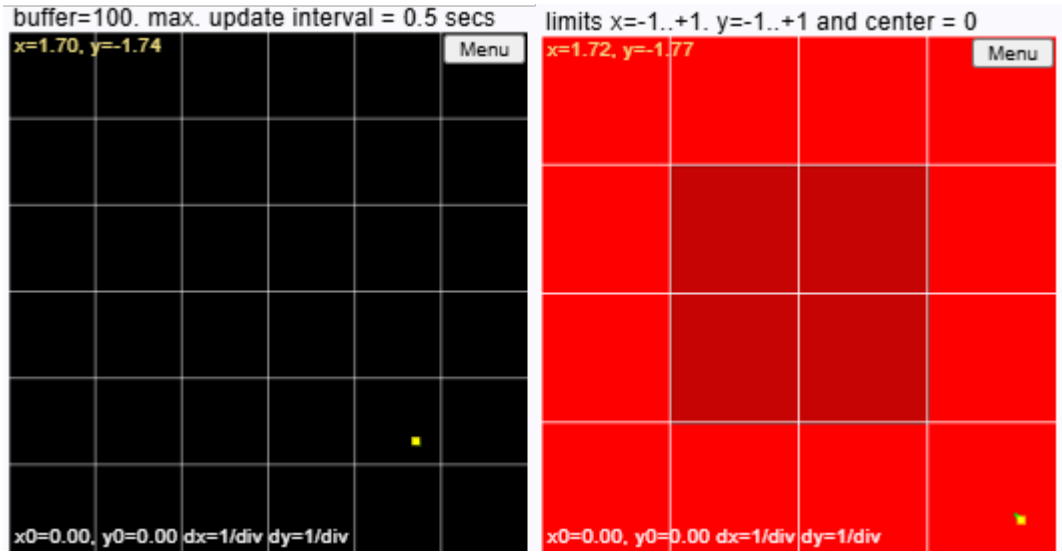
The ChartElement uses different display and scaling modes, depending on the display mode defined in *mode*. The table below lists the defined display modes an their behavior.

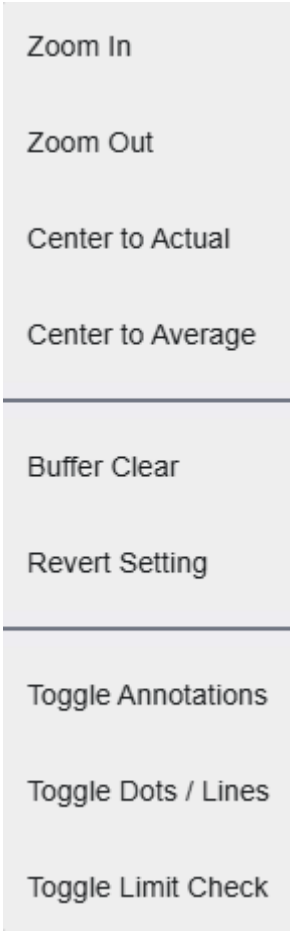
mode	**Description**
FLOATING-AUTO	Does a full autoscale. The y-offset of the strip chart is set that the newest value displayed appers at the middle of the y axis. The chart y-scale is evaluated in a 1-2-5 raster to the finest scale that allows all points in the history to be displayed in the chart area. All scaling parameters are ignored.
FLOATING-FIXED	Evaluates the y-offset like in **FLOATING-AUTO**, but applies a fixed y-scale as defined in *scale*. The *scale* value is 1/div and chart height is assumed as 2 divisions. So, the top line of the chart corresponds to y-offset + *scale*, the bottom line to y-offset -*scale*.
FIXED	Sets a fixed y-range from the *minValue* / *maxValue* parameters.

mode	**Description**
FIXED-THRESHOLD	Like **FIXED**, but also checks every new value against the *minThreshold* / *maxThreshold* limits. If outside the limits, the chart background turns to red.

2.2.2.17 XY chart element

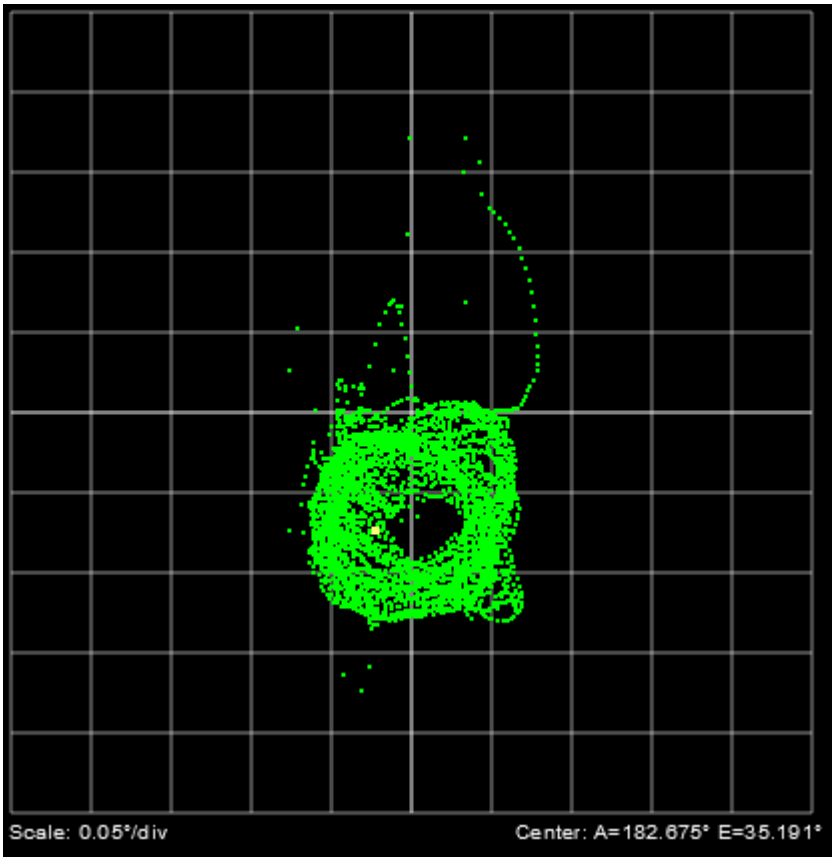
This element shows the relation of two numeric variables in an X/Y diagram, featuring a 'trace' which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

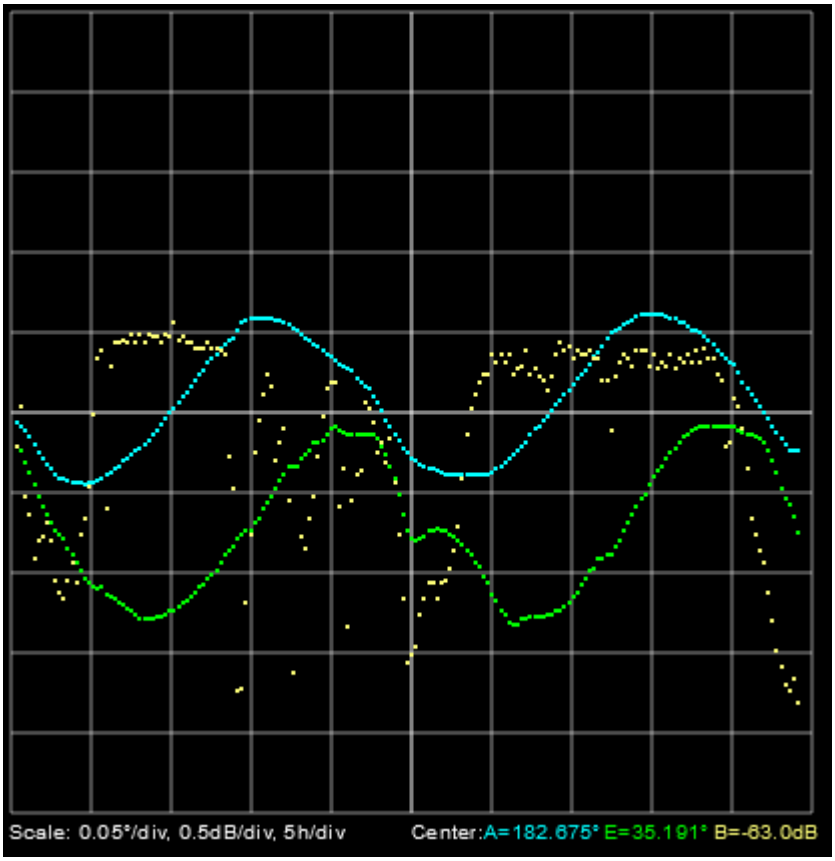




2.2.2.18 AzEl element

The AzElElement shows the tracking history of an antenna controller as a cloud of points in a elevation over azimuth coordinate system. Actually only SatService antenna controllers provide the tracking history data for this screen element.


Scale: 0.05°/div
Center: A=182.675° E=35.191°

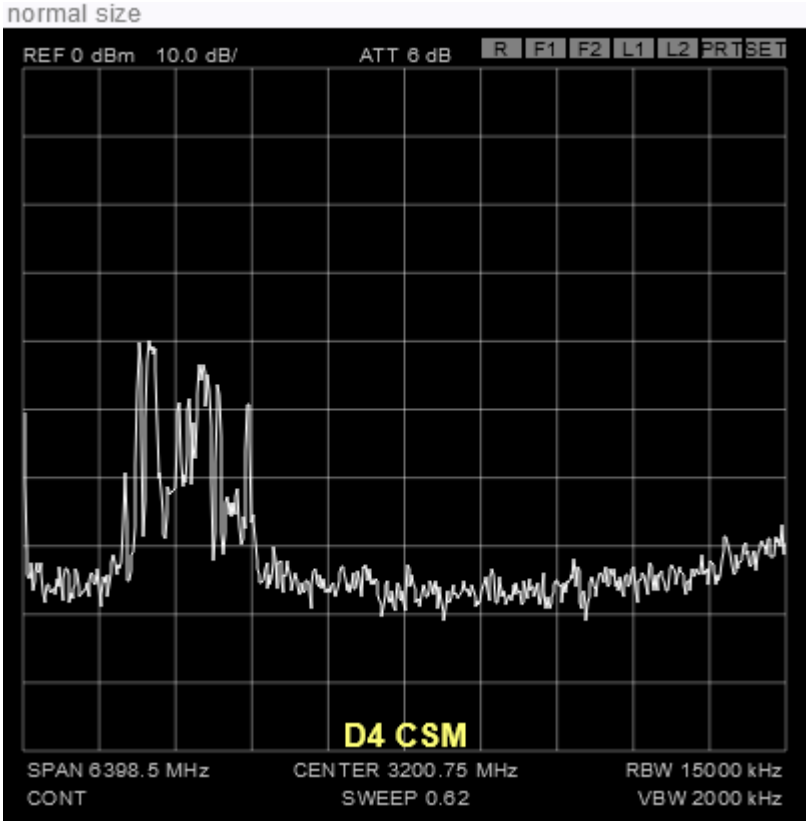

Scale: 0.05°/div, 0.5dB/div, 5h/div
Center:A=182.675° E=35.191° B=-63.0dB

2.2.2.19 Spectrum element

The spectrum element embeds the spectrum display of a spectrum analyzer device in the screen. Actually, only the CSM-Spectrum-Analyzer device may be used with the spectrum element.

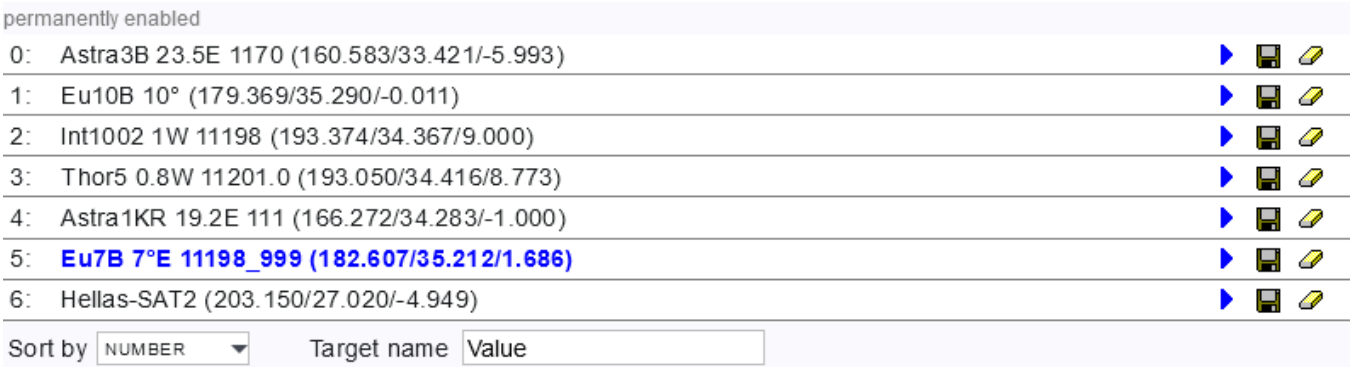
The spectrum element subscribes for a variety of parameter of the spectrum analyzer device to display its settings in the diagram area. It also permits to command some aspects of the spectrum analyzer device like the marker position. The spectrum data itself is read by subscribing for the 'trace' parameter of the device. This returns the spectrum data as a SpectrumTrace with each sweep of the spectrum analyzer. A complete description of the function of the spectrum element is given in a separate document.

normal size



2.2.2.20 Target list element

The target list element embeds the list of targets of a SatService-ACU-ODM antenna controller in the screen. Targets may be recalled (which moves the antenna to the stored position and sets the tracking parameters associated with this target), saved or deleted.



Device Name Indirection

The *id* parameter is interpreted differently depending on the context where the ODM Target List screen element resides:

- When placed in a user defined screen or in the main screen of the application, *id* is the name of the ODM device it shall refer to.
- When placed in the device screen of another device which defines a configuration variable with the antenna controller's device name, *id* is the name of this configuration variable and the name of the ODM device is derived from the content of this variable.
- Finally, when used in the device screen of the ODM device itself, *id* is '@'.

Target List Parsing

The target list element gets its information from a parameter 'target.list' the ODM device provides for this purpose. This variable contains the target list as a one line string with the target definitions appearing at fixed character positions.

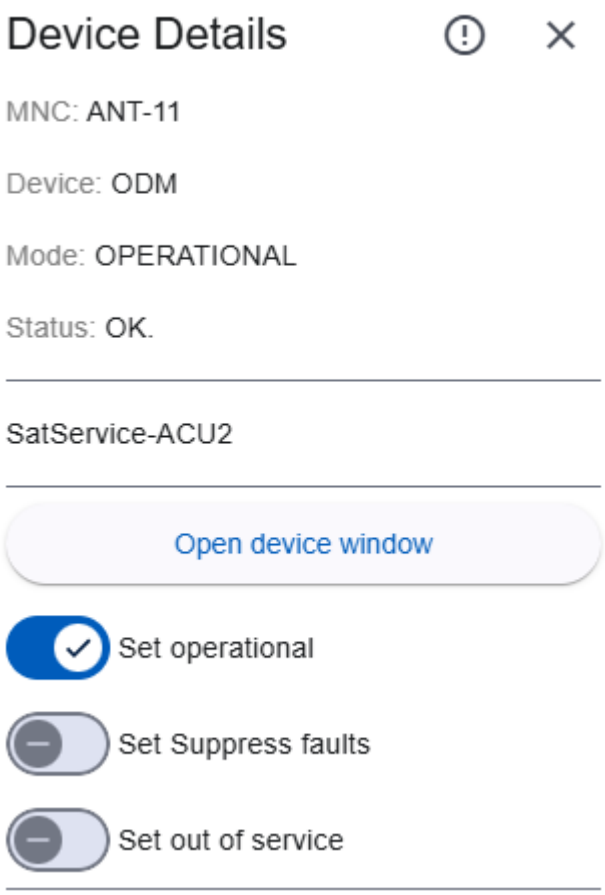
Each target definition contains the target name and the azimuth / elevation / polarization angles. The target number is defined implicitly from the position of the target in the complete string. The format of each target definition is as follows:

```
target name (azimuth/elevation/polarization)
```

This string is padded with spaces to 45 characters length. With target numbers starting at 0, you can access a particular target in the string at position *n*45* with a length of 45 characters.

2.2.3 Device details panel

The device details panel shows you the device information for a device. It is displayed on the right-hand side of the window on clicking e.g. on display element device icon. You can also perform some quick actions here, e.g. put the device into operational or out of service mode.



This panel will open if:

- click on switch icon from main screen
- clicking device in the treeview
- selecting a device from the notification panel
- selection a device from the device list page

In the title of this panel the following icons are available:

icon	description
	Click opens directly the fault page of the device. If there is any fault or warning the icon color will be red (alarm or fault) or yellow (warning) otherwise it will be gray.
	Close the device panel. You can also click outside the panel to close the panel.

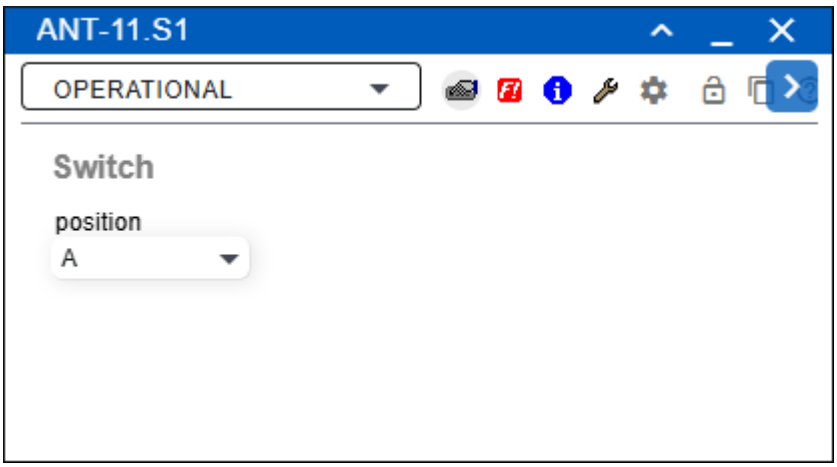
Details

- **MNC**: name of MNC where the device belongs to
- **Device**: name of the device
- **Mode**: current operations mode of the device
- **Status**: Current summary status of the device (`MNC.DEVICE.fault`)

    - **NO.**: No fault
    - **Summary FAULT**: text displayed in red color.
    - **WARNING**: text displayed in yellow color.
- **Driver name**: name of the device driver
- **Comment**: shows the device comment if set in the device config page and available for this kind of driver (`MNC.DEVICE.config.deviceComment`)
- **On Air state** shows `ON AIR` if the a devices provides this information (`MNC.DEVICE.info.signal.on`)
- **Open device window**: open device window for the device
- **Change mode switches**: change the operation mode, select between
    - OPERATIONAL: enable communcation between MNC and equipment
    - FAULT-SUPPRESSED: suppresses summary fault or warning, but stays operational
    - OUT-OF-SERVICE: disables the communication between MNC and equipment
    - MAINTENANCE: disables the communication between MNC and equipment and marks the device to be in maintenance. This triggers the inventory system if enabled.
- **Toggle switch**: If the selected device is a switch element, a toggle button is displayed. You can toggle the position of the switch `A -> B` or `B -> A`

2.3 Device window

The M&C offers a device window for each type of equipment it supports. Device windows can be launched by device details panel, device list. It shows all parameters read from the device and permits to alter each writable parameter. For clearance, the parameters are grouped to pages, you can switch between the pages using the tool-bar.



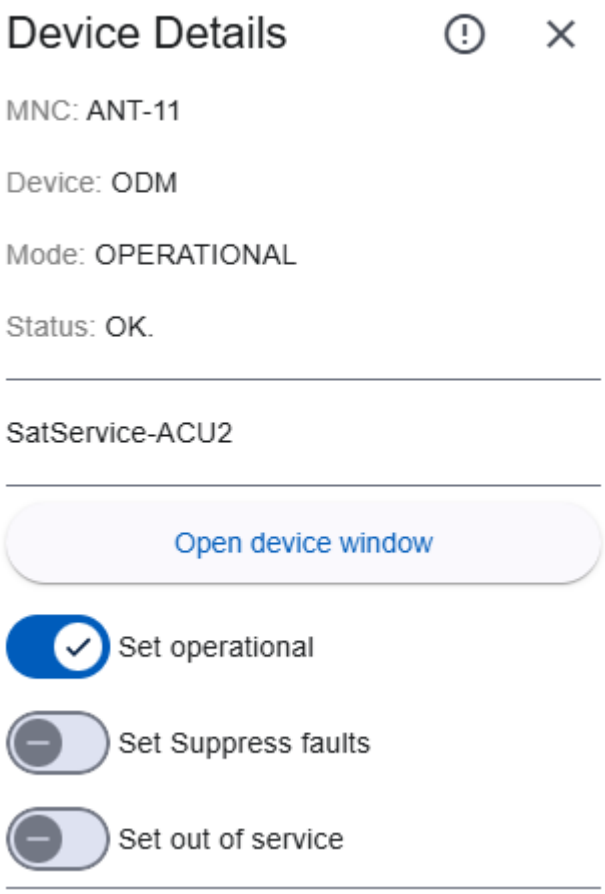
Functionality

- **Drag and drop**: Device window can be dragged from header and dropped anywhere in browser window.
- **Resize device window**: You can resize the window from every side expect top and top corners.
- **Scroll buttons**: If the toolbar menu is larger than the device window, the scroll button is displayed on the right or left side, depending on where the content is hidden.

- **Duplicate window**: The same device window can also be duplicated, which means it could be opened more the one time.
- **Debug info**: On mouse hover element details are displayed on bottom left side corner. It should be activated from 'User setting'
- **Navigates through pages**: Different type of page can be open from toolbar. Details of device window are described under 'Device window'.

2.3.1 Device details panel

The device details panel shows you the device information for a device. It is displayed on the right-hand side of the window on clicking e.g. on display element device icon. You can also perform some quick actions here, e.g. put the device into operational or out of service mode.



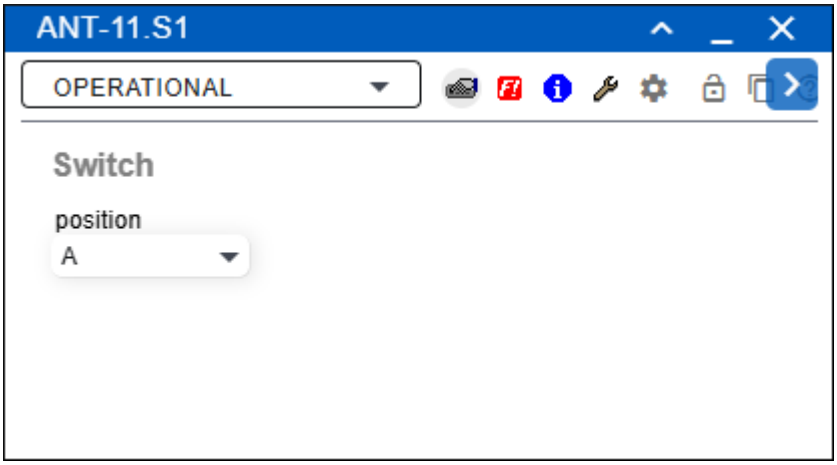
Top

- **Device details**: Title of the panel.
- **Info icon**: On click it opens a device window directly by Faults page. If there is any fault or warning the icon color will be red (alarm or fault) or yellow (warning) otherwise it will be black.
- **Close**: Close the device panel. You can also click outside the panel to close the panel.

Details

- **MNC**: The MNC name of the device.
- **Device**: The name of the device.
- **Mode**: Current mode of the device.
- **Status**: Current status (fault) of the device. "MNC.Devicename.fault" is used in this case.
  - **NO.**: No fault
  - **Summary FAULT**: Text displayed in red color.
  - **WARNING**: Text displayed in yellow color.
- **Device comment**: A device comment saved the device.
- **Open device window**: Open device window for the device.
- **Change mode switches**: By clicking on the mode set the device to the selected mode.
- **Toggle switch**: If the selected device is a display switch element, a toggle button is displayed. You can toggle the position of the switch (A -> B or B -> A)

2.3.2 Device window



The device window is a floating window, i.e. you can drag it from the header and drop it anywhere in the browser window. It has 3 main components:

Header

If the color of the header is blue, as shown in the image above, the device has no errors. If the device has an fault, the fault description (fault, warning, etc.) is displayed next to the header title and the background color of the header is also changed according to the fault (dark red on fault value "UNDEF!" and "COMFLT!", yellow on "Summary FAULT" and light red on "FAULT").

- **Title**: MNC name and device name of the device is displayed on right side of device window.
- **Roll up or down**: It roll up or down the window.
- **Minimize or Maximize window**: It minimizes the window in the bottom right-hand corner of the browser window and if window is minimized it maximizes the window to the last known position.
- **Close**: Close the device window.

Toolbar: Various devices have different options for the toolbar, but most have the common options.

- **Mode selector dropdown**: The operation mode selector displays and sets the device operation mode. Device operation modes are:
  - **OPERATIONAL**: This is the normal operation mode.
  - **FAULT-SUPPRESSED**: The M&C server normally polls and controls the device, but does not generate a device summary fault if the is a fault condition with this unit.
  - **OUT-OF-SERVICE** The M&C server does not try to communicate with the device at all. If you alter parameters of a device in this mode, this has no effect.
- : Opens a Load Device Preset dialog which lets you select a formerly stored device preset and apply the stored setting to the device. The chapter 'Preset editor' gives more information about device presets and how to handle them.
- : Click on *save* icon opens a Store Device Preset dialog which lets you store the actual device settings at the server for later retrieval.
- : Switches the window to the 'faults' page. This page displays all fault flags the software knows about this device.
- : Switches the window to the 'info' page. This page gives some information about the type of equipment controlled.
- : Switches the window to the 'maintenance' page. This page lets you set some configuration parameters and gives access to the 'low level interface' of the device.
- : Config page can be open by *settings* icon from toolbar.
- : Locks the operation of the device (through lock icon) that no other user can change device settings. See paragraph device locking below for details. If locked, the button shows a red lock.
- : Duplicates / clones this window. This is useful if you want to have more than one instance of a device window open at a time.
- **Help**: Launches the web browser with the help page for this individual device type. For a list of device types supported by the software see the 'Device Driver Reference'.

Device locking

Clicking the *settings* icon button once locks the operation of the device. The button icon changes it's color to red . If the device is locked, no other user may change devices settings. Locking is done along the user name, hence if you are logged in a second time an another place, you can operate the device from there as well.

A second click to the lock button releases the lock. Only the operator who set the lock may release it. This need not to be done at the same place where the lock was set bot you must be logged in with the same user name.

Administrators (with a privilege level of 150 or above) may release the lock in any case, also a restart of the sat-nms M&C server releases all pending device locks.

Content

Writable parameters are shown below the toolbar, read only parameters appear with the standard gray background of the window. Choice parameters are changed immediately when you alter the selection. If you edit a text parameter, the field is shown with a yellow background. Once you press ENTER or you leave the field, the changed value gets transferred

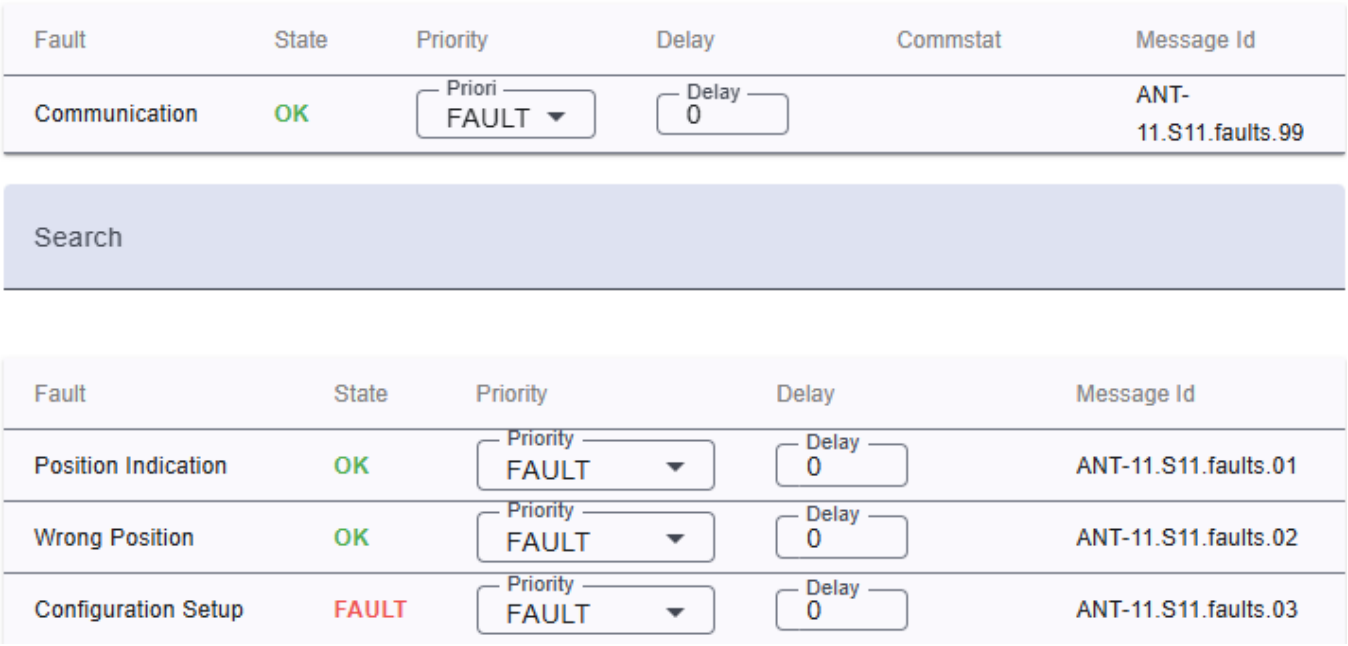
to the device. To cancel a parameter setting press ESC. The old value gets restored in the field and the background color returns to light gray. No parameter value is sent to the device in this case.

2.3.2.1 Device window pages

For a list of device types supported by the software see the'Device Driver Reference'.

- Fault page
- Info page
- Maintenance page
- Config page

2.3.2.1.1 Fault page



Moreover, you can view and change the event priority for each particular fault flag:

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

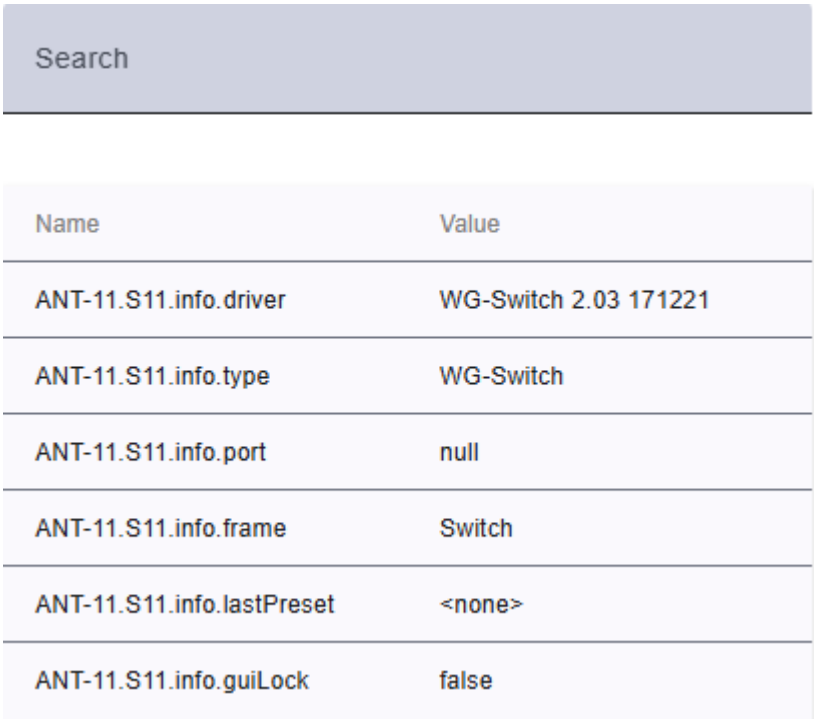
- **Fault**: The name of the fault.
- **State**: State or fault of device.
- **Priority**: Priority of the fault.
  - **OFF**: This flag is completely suppressed.

    - **INFO**: This flag is signaled as an 'informational' event.
    - **WARNING**: This flag is signaled as a 'warning' event.
    - **FAULT**: This flag is signaled as a 'fault' event.
    - **ALARM**: This flag is signaled as an 'alarm' event.
- **Delay**: Delay on trigger alarm.
- **Commstat**: (Communication status) shows if the description of fault.
- **Message id**: Message id or fault flags.

Search bar: Allows users to filter events by keywords.

2.3.2.1.2 Info page

This page gives information about the device. It is readonly page.



Search bar: Allows users to filter events by keywords.

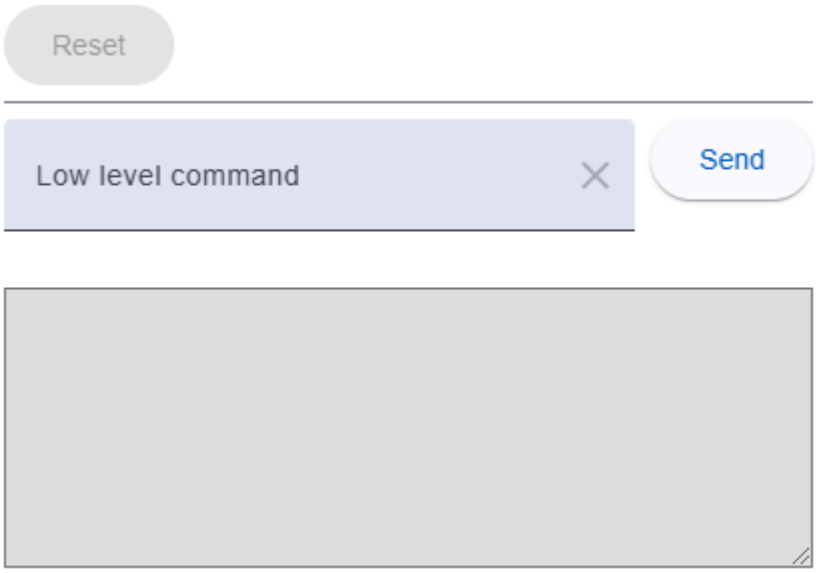
Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Name**: Message id or variables defined by this device driver.
- **Value**: Value of message id or variables defined by this device driver.

2.3.2.1.3 Maintenance page



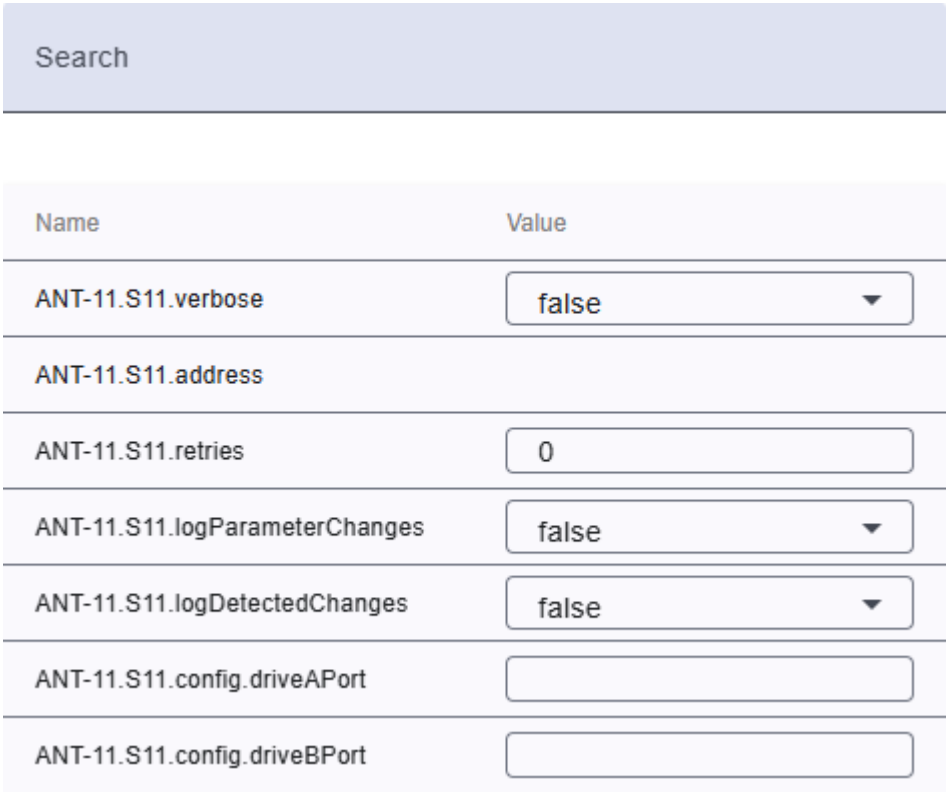
Reset: Resets the device.

Low level command: Command to check check the value.

Send: Send the low level command.

Text area: Shows the returned value of command.

2.3.2.1.4 Config page



Search bar: Allows users to filter events by keywords.

Header row

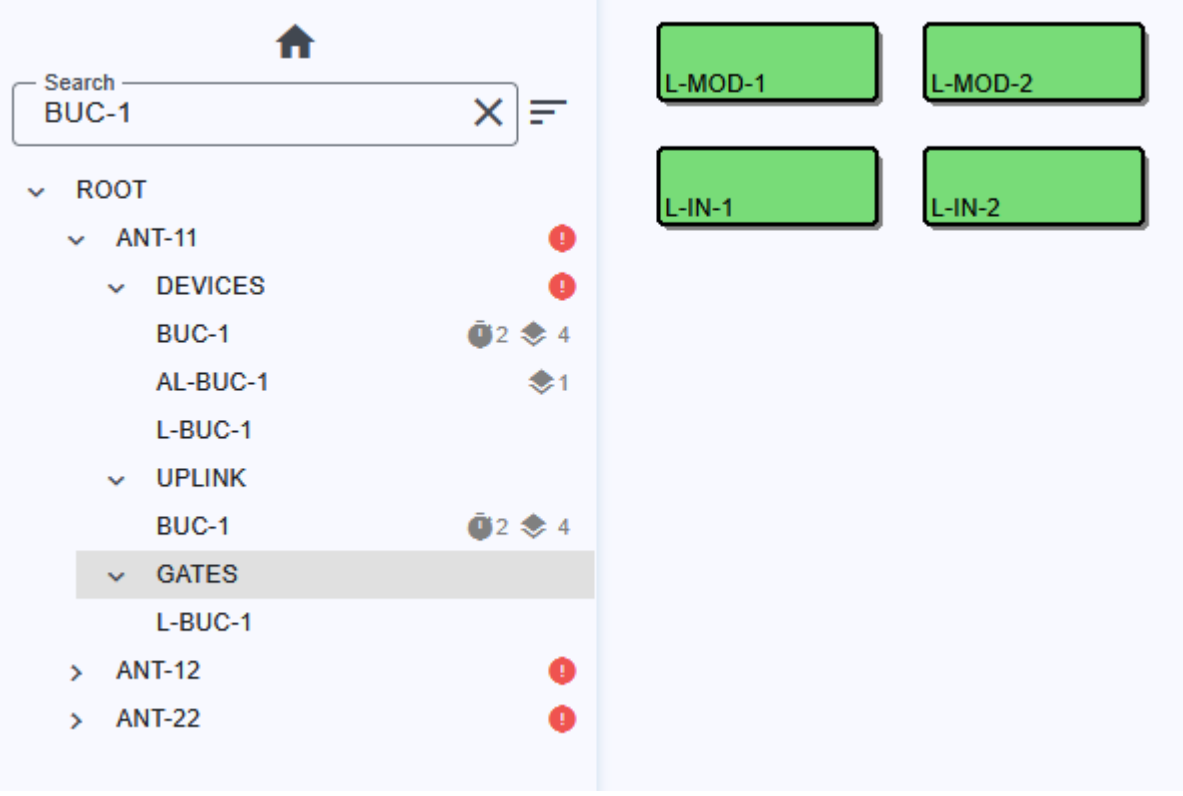
- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Name**: Configurable parameters.
- **Value**: Value of the parameters. The values can be changed by clicking on input fields. After you have changed the value in the input fields, simply press the `enter` key or click with the mouse outside the input field to send the values. Press the `esc` key to discard or undo changes. It works only with input fields not with dropdowns

2.4 Treeview

The so call Treeview provides a list of devices of all connected MNC system organized in a tree. It can be accessed from the sidebar *tree icon* below the logo. You can arrange the devices and sub-trees in the Treeview Editor



- **Home icon**: Hide the treeview.
- **Search**: filter devices by name
- **Sort**: Tree can be sorted by

    - ascending device name
    - descending device name
    - default: as definded in TreeView Editor (natural order)

The tree shows all connected MNC's as top node and the configured subsystems. Each node can have one or more devices attached.

Each device node is marked with its fault state with icons

- no icon: ok
- grey icon: state cannot be determined
- yellow icon: device summary state is warning
- red icon: device summary state is fault or alarm

Subsystem nodes inherit the fault state from the subordinate elements, summed up from the devices and subsystems.

Devices with **masked** or **delayed** faults will additionally show the count of these faults.

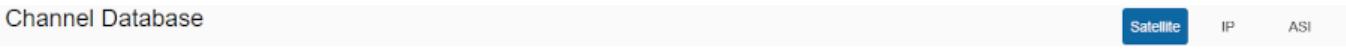
Functionality

- **>** : Expand the node.
- **<** : Collapse the node.
- **Click on node**: expands the node and will be load configured screen. If no screen for this subsystem is defined it shows device icons of all devices which belongs to the subsystem.
- **Click on device**: Opens device details panel on right hand side

2.5 Channel database

The Channel Database is a central store for receive and transmit channels. These settings are used to configure equipment for receiving or transmitting to a satellite or on an IP network (streaming).

The title bar provides access to the three different channel types:



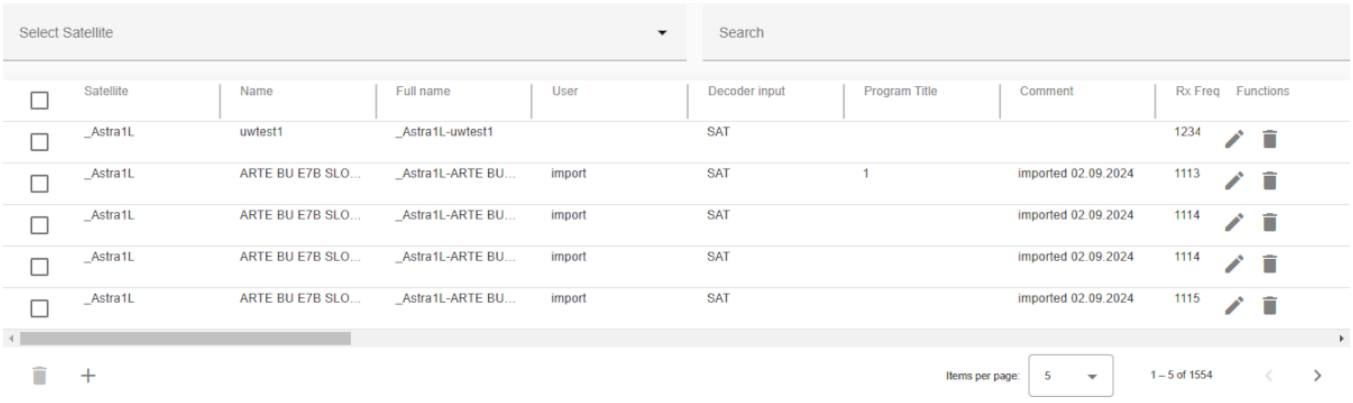
- Satellite channels (default)
    - satellite channels include parameters like frequency, modulation type
    - transmit and receiving via satellite is supported
    - basic audio/video settings are also part of these channels
- IP/Streaming channels
    - IP channels support various network streaming protocols like UDP or RTP
    - transmit and receiving via network is supported
    - basic audio/video settings are also part of these channels
- ASI channels
    - basically ASI channels are only to address the ASI inputs of IRDs
    - there are no specific settings on the ASI interface needed
    - basic audio/video settings are also part of these channels

You can choose which type of channels you want by clicking on the buttons at the top of this page.

On a system without installed SatDB or if the PostgreSQL Database is not reachable an error message will be display: Error DB not reachable. In this case, please check your database installation.

2.5.1 Satellite channels

The list of all satellite channels in the database. The table can be scrolled horizontally. Satellite and functions row are sticky.



Satellite channels define the common paramaters of receive equipment (e.g. IRDs, Demodulator) and transmit equipment (e.g. Modulators) for satellite transmission. Typically used for the contribution of newsfeeds or TV over satellite but not limited to. Each channel belongs to a satellite which needs to be configured in the Satellite Database.

To use this database you have to configure the corresponding Channel-Devices on the MNC server(s) which controlls the equipment.

For receive applications the following logical devices are available:

- RX-Channel-ByChannel
- RX-Channel-BySatellite
- RX-Channel-ActivationBased

For transmit applications the following logical devices are available:

- TX-Channel-ByChannel
- TX-Channel-BySatellite

These logical devices will retrieve the channel information directly from the central database. Please refer to the linked documentation for details how to setup and use the RX and TX Channel devices.

Search bars

- Located at the top of the table.

- Allows users to filter by:
  - *Satellite*: select one satellite from the list to display all related channels
  - *Keywords*: full-text search over all channels and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.
- You can arrange the width of an row by `click + hold + drag` near from the header name (grey border)

Table columns

The first column located at the far left of the table provides a checkbox to select one or more channels (rows) and perform the same operation on all selected channels.

- **Single Selection**: Click on a single checkbox to select an individual channel
- **Multiple Selection**: Click multiple checkboxes to select multiple channels.
- **Select all**: The checkbox in the header row allows you to select or deselect all channels on all pages.

The following function(s) are available:

Function	Icon	Description
delete		delete the selected channels from the database

The last column located at the far right of the tables provides buttons to run actions on the corresponding channel.

The following functions for managing a channel are `available` :

Function	Icon	Description
Delete		to delete the channel
Edit		modify/edit channel

The following buttons for global function(s) are available at the the bottom of the table:

Function	Icon	Description
Add		add a new channel

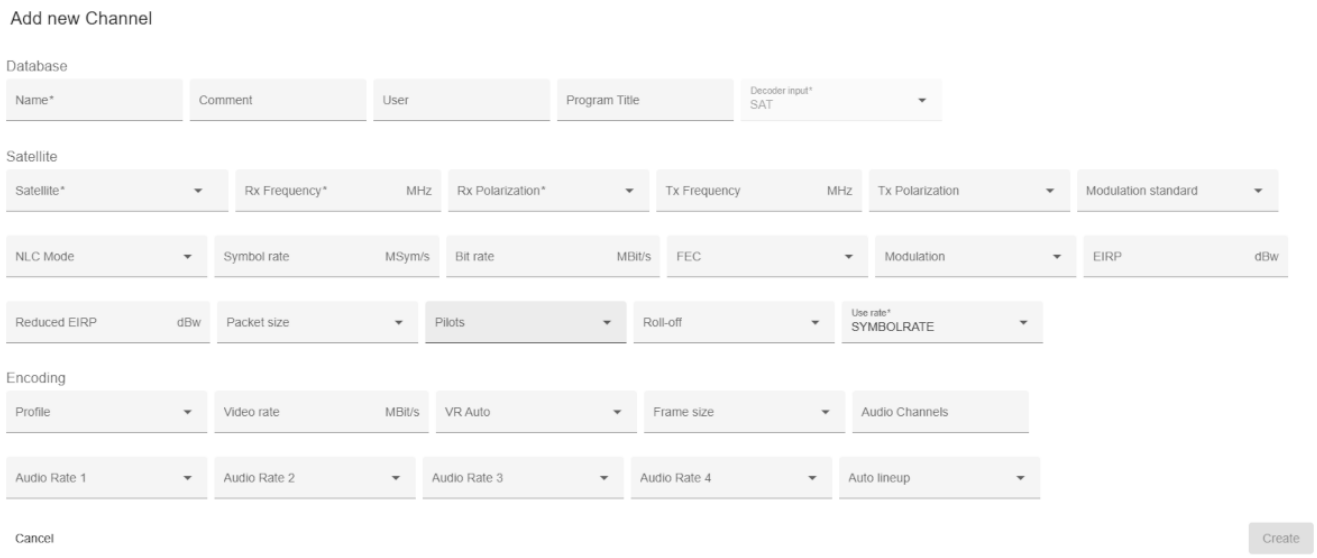
Function	Icon	Description
Delete		to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The satellite channel table contains the following columns:

Column name	Description
Satellite	name of satellite
Name	user defined channel name
Full name	automatically generated full name (`satellite's name` + `-` + `Name`)
User	free text to enter a user name which is responsible for this database entry
Program title	name of the program transmitted or received with this channel
Comment	one line comment describing this satellite channel
Rx frequency	receive frequency in MHz.
Rx pol.	receive polarization
Tx frequency	transmit frequency in MHz
Tx pol.	transmit polarization
Mode	modulation standard
NLC mode	NLC (Non Linear Constellation) mode for NS4 modulation
Symbol rate	symbol rate in MSym/sec
FEC	FEC rate
Modulation	modulation type
Roll-off	roll off factor
Pilots	pilots on/off switch
Bit rate	data rate in MBit/sec
Use rate	selects the way the data rate is specified

Column name	Description
Packet size	packet size
Frame size	frame size
Video rate	video data rate in MBit/sec
VR auto	video rate auto switch
Profile	video encoding profile
EIRP	transmit EIRP in dBW
Reduced EIRP	reduced EIRP for line up in dBW
Auto lineup	configures automatic line up
Audio rate 1	audio channel 1 data rate
Audio rate 2	audio channel 2 data rate
Audio rate 3	audio channel 3 data rate
Audio rate 4	audio channel 4 data rate
Audio channels	Number of audio channels.

2.5.1.1 Add/Edit satellite channel dialog



The Add or Edit satellite channel dialog shows up on

- click on button on the bottom toolbar
- click on in the channel row

The dialog title clearly indicates whether you are creating a new channel or editing an existing one, ensuring clarity in the action being performed.

Required fields are marked with an asterisk `*`. Fields which you don't plan to use in your installation (e.g. TX settings if you only receive channels) can be can remain empty or on defaults. Certain setting will anyway only be applied to equipment if this is configured in the RX/TX-Channel devices.

Field	Description
Name*	user defined channel name, must be unique
Comment	describe this satellite channel
User	free text to enter a user name which is responsible for this database entry, must not match to a username of this M&C system
Program title	name of the program transmitted or received with this channel
Decoder input	*read-only* channel type
Satellite*	select a satellite name to which this channel belongs to
Rx frequency*	receive frequency in MHz (3 digits precision)
Rx polarization*	receive polarization, one of `X` (Horizontal), `Y` (Vertical), `L` (LHCP) or `R` (RHCP)
Tx frequency	transmit frequency in MHz (3 digits precision).
Tx polarization	transmit polarization, one of `X` (Horizontal), `Y` (Vertical), `L` (LHCP) or `R` (RHCP)
Modulation standard	DVB mode should be one of the option from the drop-down list.
NLC mode	enable (`ON`) or disable (`OFF`) NLC mode for NS4 modulations
Symbol rate	symbol rate in MSym/sec (4 digits precision)
Bit rate	data rate in MBit/sec (4 digits precision)
FEC	select FEC rate from drop-down list

Field	Description
Modulation	select modulation from drop-down list
EIRP	transmit EIRP in dBW (2 digits precision)
Reduced EIRP	reduced EIRP for line-up in dBW (2 digits precision)
Packet size	packet size, one of `188` or `204`
Pilots	enable (`ON`) or disable (`OFF`) pilots
Roll-off	select roll off factor from drop-down list
Use rate	selects the way the data rate is specified: by `SYMBOLRATE` (default) or by `BITRATE` . Make sure that the corresponding rate is also stored with this channel.
Profile	select video encoding profile from drop-down list
Video rate	video data rate in MBit/sec (4 digits precision)
VR auto	video rate auto switch, one of `FIXED` or `AUTO`
Frame size	frame size, one of `SHORT` or `NORMAL`
Audio channels	number of audio channels between 1-4
Audio rate 1	audio channel 1 data rate select from drop-down list
Audio rate 2	audio channel 2 data rate select from drop-down list
Audio rate 3	audio channel 3 data rate select from drop-down list
Audio rate 4	audio channel 4 data rate select from drop-down list
Auto lineup	enable (`ON`) or disable (`OFF`) automatic line up

The following actions are `available` :

Function	Description
Create	For new channel, this creates the channel with the provided details
Save	For existing channel, this saves modifications.
Cancel	Closes the dialog without saving

The form tests the input fields for validity and display hints in red in each field with wrong

input. If an error occurs in the input fields, the button Create/Save button is deactivated.

2.5.1.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

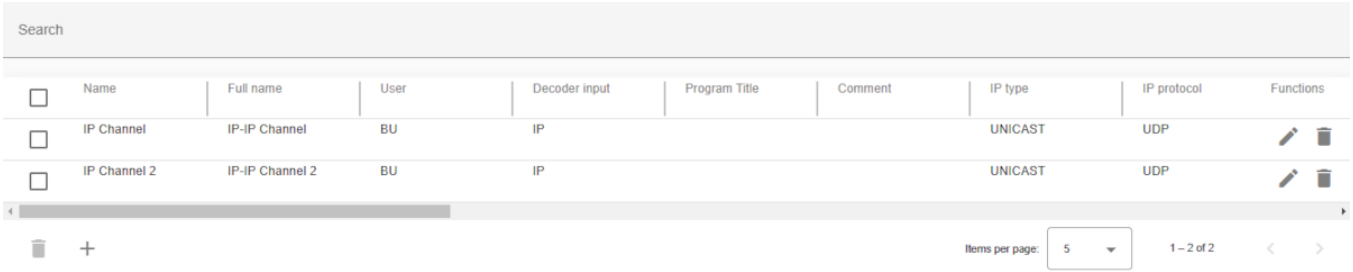
- click on in the table row on the bottom toolbar

The following actions are `available` :

Function	Description
Yes	Deletes the channel(s) permanently
Cancel	Cancels the deletion action

2.5.2 IP channels

The list of all IP / Streaming channels in the database. The table can be scrolled horizontally. Name and functions row are sticky.



IP channels define the common parameters for receiving and transmitting IP streams. Typically used in news contribution over Internet but not limited to. Each channel contains source and/or destination addresses and information of the used streaming protocol.

To use this database you have to configure the corresponding Channel-Devices on the MNC server(s) which controlls the equipment.

For receive applications the following logical devices are available:

- RX-Channel-ByChannel
- RX-Channel-BySatellite
- RX-Channel-ActivationBased

For transmit applications the following logical devices are available:

- TX-Channel-ByChannel
- TX-Channel-BySatellite

These logical devices will retrieve the channel information directly from the central database.

Please refer to the linked documentation for details how to setup and use the RX and TX Channel devices.

Search bars

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all channels and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.
- You can arrange the width of an row by `click + hold + drag` near from the header name (grey border)

Table columns

The first column located at the far left of the table provides a checkbox to select one or more channels (rows) and perform the same operation on all selected channels.

- **Single Selection**: Click on a single checkbox to select an individual channel
- **Multiple Selection**: Click multiple checkboxes to select multiple channels.
- **Select all**: The checkbox in the header row allows you to select or deselect all channels on all pages.

The following function(s) are available:

Function	Icon	Description
delete		delete the selected channels from the database

The last column located at the far right of the tables provides buttons to run actions on the corresponding channel.

The following functions for managing a channel are `available` :

Function	Icon	Description
Delete		to delete the channel
Edit		modify/edit channel

The following buttons for global function(s) are available at the the bottom of the table:

Function	Icon	Description

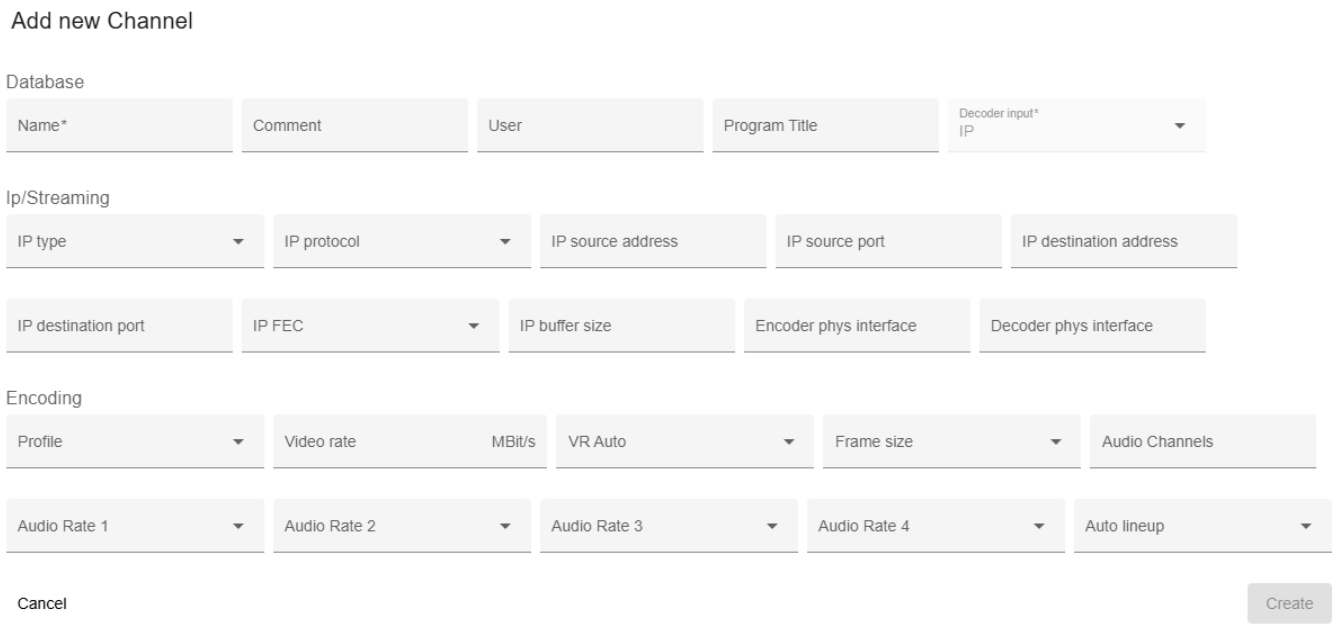
Function	Icon	Description
Add	+	add a new channel
Delete	🗑	to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The IP channel table contains the following columns:

Column name	Description
Name	user defined channel name
Full name	automatically generated full name (IP + - + Name)
User	free text to enter a user name which is responsible for this database entry
Program title	name of the program transmitted or received with this channel
Comment	one line comment describing this satellite channel
IP type	IP addressing type
IP protocol	IP streaming protocol
IP source address	Source IP address
IP source port	Source port number
IP destination address	Destination IP address
IP destination port	Destination port number
IP FEC	IP FEC selection
IP buffer size	Buffer size
Encoder phys.int	Encoder physical output number
Decoder phys.int.	Decoder physical input number
Frame size	The frame size
Video rate	video data rate in MBit/sec
VR auto	video rate auto switch
Profile	video encoding profile

Column name	Description
Auto lineup	configures automatic line up
Audio rate 1	audio channel 1 data rate
Audio rate 2	audio channel 2 data rate
Audio rate 3	audio channel 3 data rate
Audio rate 4	audio channel 4 data rate
Audio channels	Number of audio channels.

2.5.2.1 Add/Edit IP channel dialog**



The Add or Edit IP channel dialog shows up on

- click on + button on the bottom toolbar
- click on ✎ in the channel row

The dialog title clearly indicates whether you are creating a new channel or editing an existing one, ensuring clarity in the action being performed.

Required fields are marked with an asterisk `*` . Fields which you don't plan to use in your installation (e.g. TX settings if you only receive channels) can be can remain empty or on defaults. Certain setting will anyway only be applied to equipment if this is configured in the RX/TX-Channel devices.

Field	Description

Field	Description
Name*	user defined channel name, must be unique
Comment	describe this IP channel
User	free text to enter a user name which is responsible for this database entry, must not match to a username of this M&C system
Program title	name of the program transmitted or received with this channel.
Decoder input	*read-only* channel type
IP type	IP addressing type, one of `UNICAST` or `MULTICAST`
IP protocol	IP protocol, one of `UDP`, `RTP`, `TCP` or `ZIXI`
IP source address	source IP address in dotted quad notation
IP source port	source port number `0-65535`
IP destination address	destination IP address in dotted quad notation
IP destination port	destination port number `0-65535`
IP FEC	IP FEC selection, one of `OFF` or `ON`
IP buffer size port	buffer size
Encoder phys.int.	encoder physical output number (1-4)
Decoder phys.int.	decoder physical input number (1-4)
Profile	select video encoding profile from drop-down list
Video rate	video data rate in MBit/sec (4 digits precision)
VR auto	video rate auto switch, one of `FIXED` or `AUTO`
Frame size	frame size, one of `SHORT` or `NORMAL`

Field	Description
Audio channels	number of audio channels should be between 1-4
Audio rate 1	audio channel 1 data rate select from drop-down list
Audio rate 2	audio channel 2 data rate select from drop-down list
Audio rate 3	audio channel 3 data rate select from drop-down list
Audio rate 4	audio channel 4 data rate select from drop-down list

The following actions are `available` :

Function	Description
Create	For new channel, this creates the channel with the provided details
Save	For existing channel, this saves modifications.
Cancel	Closes the dialog without saving

The form tests the input fields for validity and display hints in red in each field with wrong input. If an error occurs in the input fields, the button Create/Save button is deactivated.

2.5.2.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

- click on in the table row on the bottom toolbar

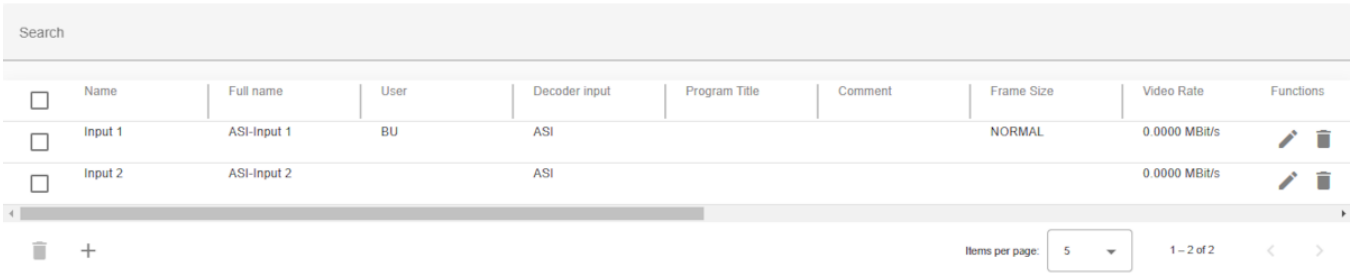
The following actions are `available` :

Function	Description
Yes	Deletes the channel(s) permanently
Cancel	Cancels the deletion action

2.5.3 ASI channels

The list of all ASI channel records in the database. The table can be scrolled horizontally. Name

and functions row are sticky.



Some receives (e.g. IRDs) provides between Satellite- and IP-Inputs also plain ASI inputs. In this mode no configuration of the input is required, you just have to switch to ASI input. Therefore the related channels are used in most cases as placeholder and to setup encoding (audio,video settings) in case that the receiver get a signal to ASI input. Nevertheless these channel type is available to allow a consistent operation with all input methods.

For receive applications the following logical devices are available:

- RX-Channel-ByChannel
- RX-Channel-BySatellite
- RX-Channel-ActivationBased

For transmit applications the following logical devices are available:

- TX-Channel-ByChannel
- TX-Channel-BySatellite

These logical devices will retrieve the channel information directly from the central database. Please refer to the linked documentation for details how to setup and use the RX and TX Channel devices.

Search bars

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all channels and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.
- You can arrange the width of an row by `click + hold + drag` near from the header name (grey border)

Table columns

The first column located at the far left of the table provides a checkbox to select one or more channels (rows) and perform the same operation on all selected channels.

- **Single Selection**: Click on a single checkbox to select an individual channel
- **Multiple Selection**: Click multiple checkboxes to select multiple channels.
- **Select all**: The checkbox in the header row allows you to select or deselect all channels on all pages.

Function	Icon	Description
delete		delete the selected channels from the database

The last column located at the far right of the tables provides buttons to run actions on the corresponding channel.

The following functions for managing a channel are `available` :

Function	Icon	Description
Delete		to delete the channel
Edit		modify/edit channel

The following buttons for global function(s) are available at the the bottom of the table:

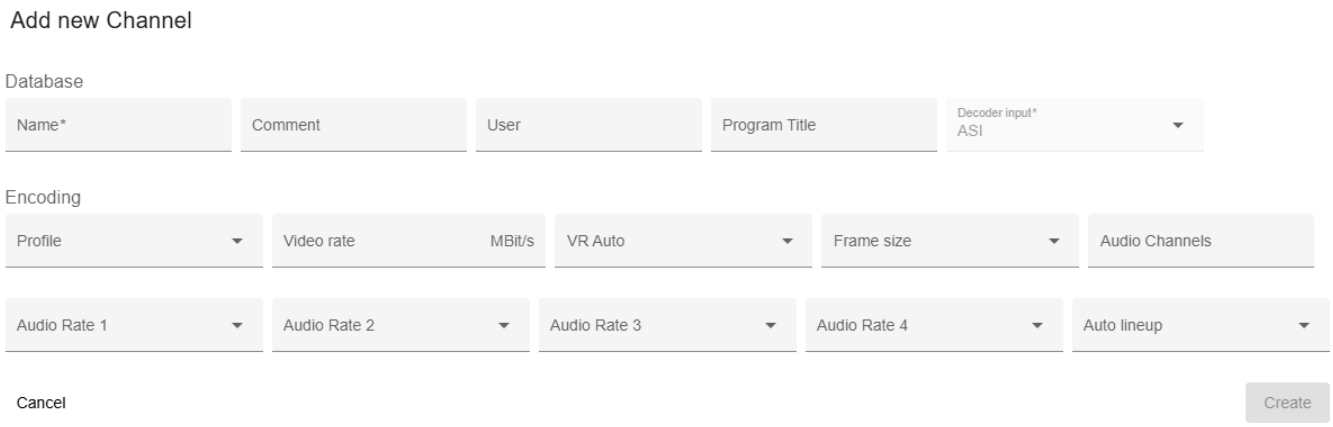
Function	Icon	Description
Add		add a new channel
Delete		to delete all selected the channels
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The ASI channel table contains the following columns:

Column name	Description
Name	user defined channel name
Fullname	automatically generated full name (`IP` + `-` + `Name`)
User	name of the user who lastly edited the channel
Program title	name of the program transmitted or received with this channel
Comment	one line comment describing this satellite channel
Frame size	frame size

Column name	Description
Video rate	video data rate in MBit/sec
VR auto	video rate auto switch
Profile	video encoding profile
Audio channels	number of audio channels between 1-4
Audio rate 1	audio channel 1 data rate select from drop-down list
Audio rate 2	audio channel 2 data rate select from drop-down list
Audio rate 3	audio channel 3 data rate select from drop-down list
Audio rate 4	audio channel 4 data rate select from drop-down list

2.5.3.1 Add/edit ASI channel dialog**



The Add or Edit ASI channel dialog shows up on

- click on + button on the bottom toolbar
- click on ✎ in the channel row

The dialog title clearly indicates whether you are creating a new channel or editing an existing one, ensuring clarity in the action being performed.

Required fields are marked with an asterisk `*`. Fields which you don't plan to use in your installation (e.g. TX settings if you only receive channels) can be can remain empty or on defaults. Certain setting will anyway only be applied to equipment if this is configured in the RX/TX-Channel devices.

Field	Description
Name*	user defined channel name, must be unique

Field	Description
Comment	describe this IP channel
User	free text to enter a user name which is responsible for this database entry, must not match to a username of this M&C system
Program title	name of the program transmitted or received with this channel.
Decoder input	*read-only* channel type
Profile	select video encoding profile from drop-down list
Video rate	video data rate in MBit/sec (4 digits precision)
VR auto	video rate auto switch, one of `FIXED` or `AUTO`
Frame size	frame size, one of `SHORT` or `NORMAL`
Audio channels	number of audio channels should be between 1-4
Audio rate 1	audio channel 1 data rate select from drop-down list
Audio rate 2	audio channel 2 data rate select from drop-down list
Audio rate 3	audio channel 3 data rate select from drop-down list
Audio rate 4	audio channel 4 data rate select from drop-down list

The following actions are `available` :

Function	Description
Create	For new channel, this creates the channel with the provided details
Save	For existing channel, this saves modifications.
Cancel	Closes the dialog without saving

The form tests the input fields for validity and display hints in red in each field with wrong

input. If an error occurs in the input fields, the button Create/Save button is deactivated.

2.5.3.2 On delete confirmation dialog

The Delete confirmation dialog shows up on

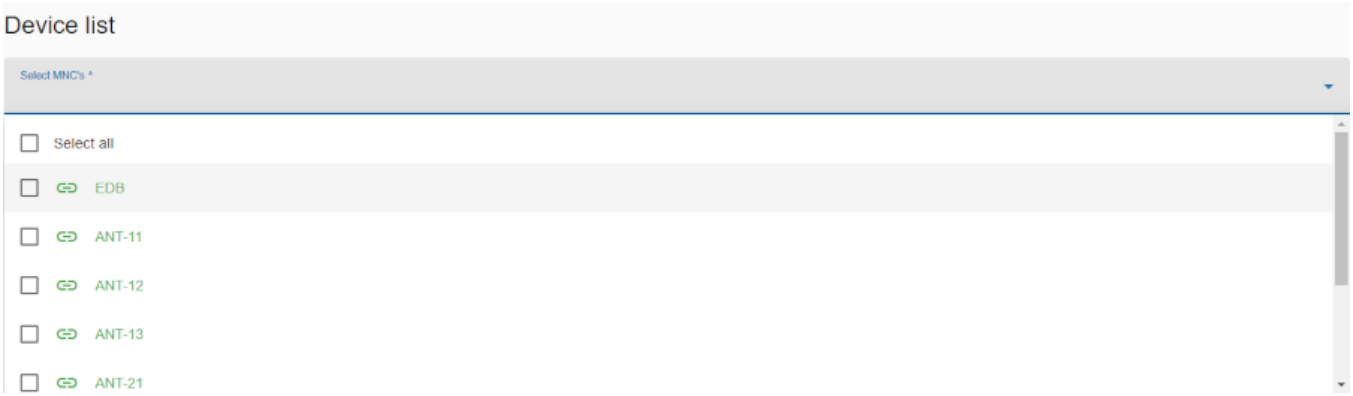
- click on in the table row on the bottom toolbar

The following actions are `available` :

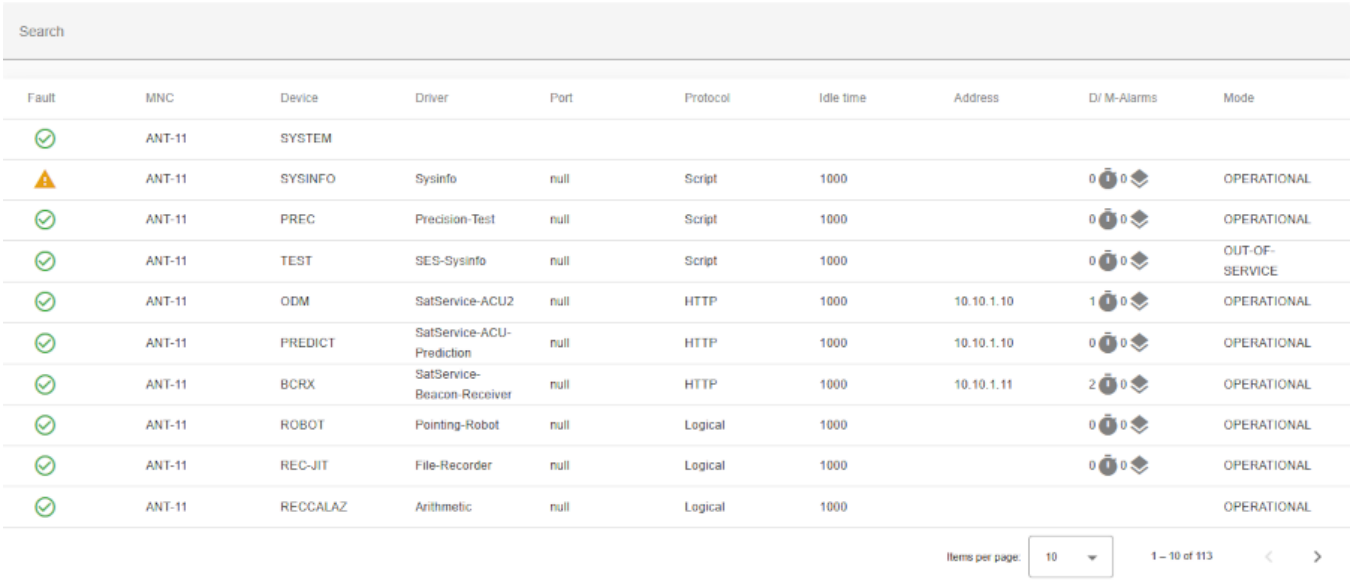
Function	Description
Yes	Deletes the channel(s) permanently
Cancel	Cancels the deletion action

2.6 Device list

The device list page shows different type of information about installed devices. For example driver name, protocol, mode or fault state. You can choose one or multiple MNCs to search and list devices.



In the first step select one or more MNCs from the drop down list below the heading. The page will show a table of alle devices.



Search bar

- Located at the top of the table.
- Allows users to filter device list
- The search pattern will be used for a full text search in all columns of the table.

Header row

- You can sort rows in ascending or descending order by clicking on the column header (e.g. MNC, device, etc.).
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is indicated by an arrow or icon near the column name.

Table columns

- **Fault**: Current fault of device on MNC. If there is no fault, a green*check icon* appears, if there is a warning, an orange *warning icon* appears and if there is a summary fault, a red *error icon* appears.
  - **Functionality**: On mouse hover on fault icons gives you detailed information.
- **MNC**: MNC name.
- **Device**: Device name.
- **Driver**: Device driver.
- **Port**: The communication port/interface.
- **Protocol**: The used protocol type.
- **Idle time**: The idle time in milliseconds.
- **Address**: Address.
- **D/M-alarms**: Delayed alarms are labelled with a clock icon and masked alarm with layers icon
- **Mode**: Current mode of device.
  - **Functionality**: On click on mode row, open device details panel.

Bottom toolbar

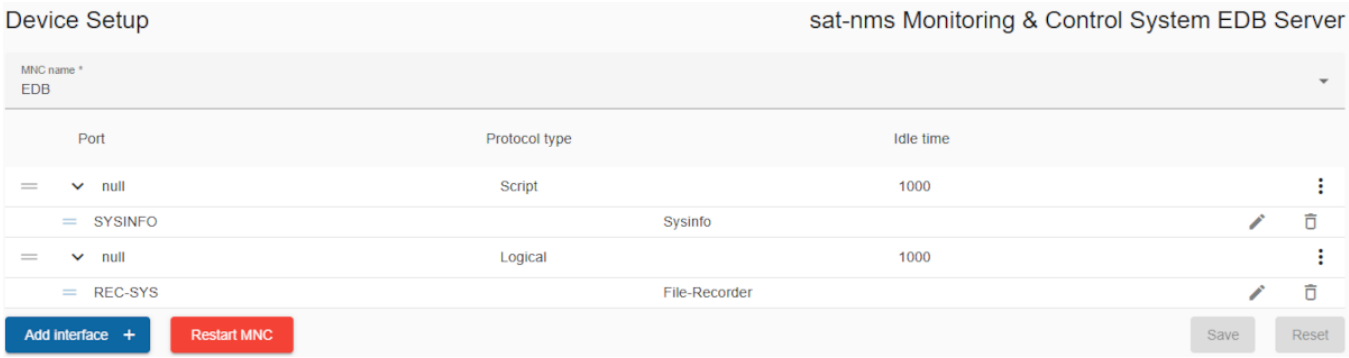
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

Functionality

- **Open device window**: On click on row expect on mode column opensdevice window.
- **Open device panel**: On click on mode row, opendevice details panel.

2.7 Device setup

With the ***sat-nms*** M&C software, the assignment which type of device is controlled by which communication interface and which driver is free configurable. The user interface of the software provides a configuration window which enables the authorized operator to setup devices.



When an operator is going to configure an MNC first it is important to understand, that with the ***sat-nms*** software each interface is bound to a communication protocol. The communication protocol used for a certain device is not unchangeably linked to the device driver. This is to support multiple devices being controlled via a single interface. Devices using a common 'multi drop' capable protocol may be connected to the MNC with a serial but like RS485.

Once an interface (serial port or network) has been included into the setup and got a communication protocol assigned to it, within the setup devices can be attached to this interface. With the device definition, the device gets a name and a device driver assigned to it. The device's name acts as an address within the software where to send at commands (equipment settings) to, the device driver tells the software how to treat the device at a low communication level.

To configure device first select the MNC:

MNC name: Select an MNC to load devices and start editing.

This will show a list of all already configured interface and their attached devices.

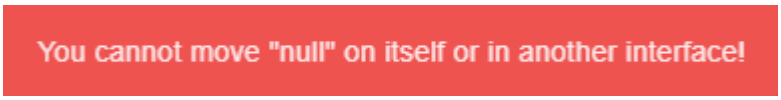
The first level of this tree represents the interface.

- **Port**: Displays the name of the port or node.

- **Protocol** Type: Indicates the protocol associated with the entry.
- **Idle Time**: Shows the idle time for the specific entry (in milliseconds).

One or more devices can be attached to one interface.

- **Parent node/Interface** These are the primary entries that may have child nodes/devices nested under them. Parent nodes or interfaces are identified by a dropdown arrow (▼) on the left side of the row, allowing you to expand or collapse their child nodes.
- **Drag Handles**: Located on the left of each row (indicated by the three horizontal lines). Users can drag and drop rows to rearrange the list dynamically. But parent nodes/interfaces cannot be moved into another interface. At the bottom of the page, the snack bar (error) would be displayed with the description of the error, as shown below.



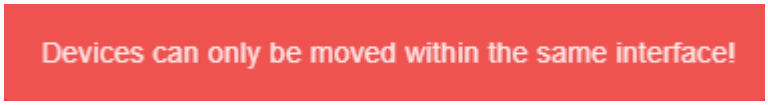
Expandable/Collapsible Rows = Interfaces

Rows with sub-items have a toggle arrow on the left side. Clicking the arrow expands or collapses the child node or devices.

- **Context Menu**: Accessible via the three-dot icon on the right of each row, provides the following options:
  - **Add Interface Below**: Inserts a new interface directly below the selected interface.
  - **Add Device**: Adds a new device as a child node under the selected interface.
  - **Edit Interface**: Opens a dialog to modify the details of the selected interface.
  - **Delete Interface**: Removes the selected interface and all its devices from the list.

Child Nodes = Devices These are nested entries directly associated with a parent node or interface. Child nodes or devices are indented beneath their respective parent node or interface to visually indicate the hierarchy.

- **Drag Handles**: Located on the left of each row (indicated by the three horizontal lines). Users can drag and drop rows to rearrange the list dynamically. However, devices cannot be moved from one interface to another interface. At the bottom of the page, the snack bar (error) would be displayed with the description of the error, as shown below.

- **Device name**: Name of the near the drag handel
- **Device driver**: Next to device name is device driver.
- **Edit device**: If you click on the *pencil icon* on the right-hand side, you can edit the device details.
- **Delete device**: Click the *trash icon* on the right hand side to delete an device from list.

Functions

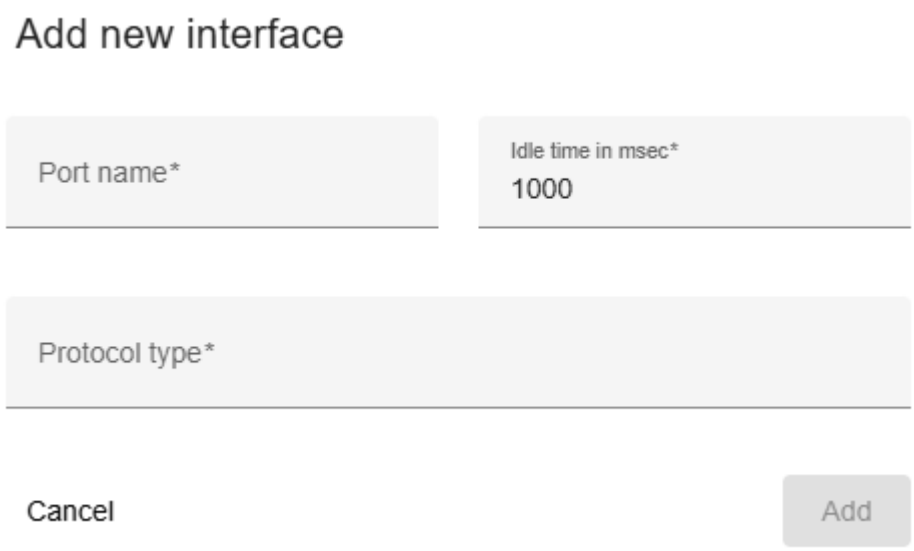
- **Add interface**: Found at the bottom left of the editor. Clicking this opens a dialog to add

a new interface to the list.

- **Restart MNC**: Clicking this button will restarts the selected MNC. Of course only after your confirmation via dialog. You must do this, to make the target system use a modified equipment setup.
- **Save**: Saves all changes made in the editor. If you have not saved the changes and accidentally leave the editor, a confirmation dialog will appear in which you must give your consent to leave the editor/page.
- **Reset**: Discards all unsaved changes and reverts to the first loading state.

2.7.1 Dialogs

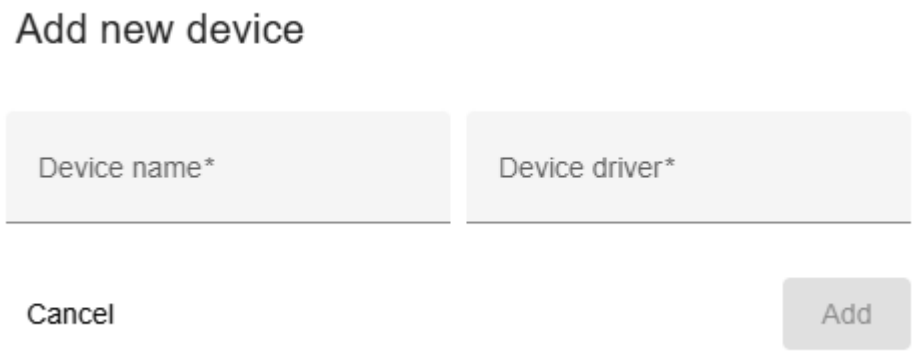
Add/edit interface



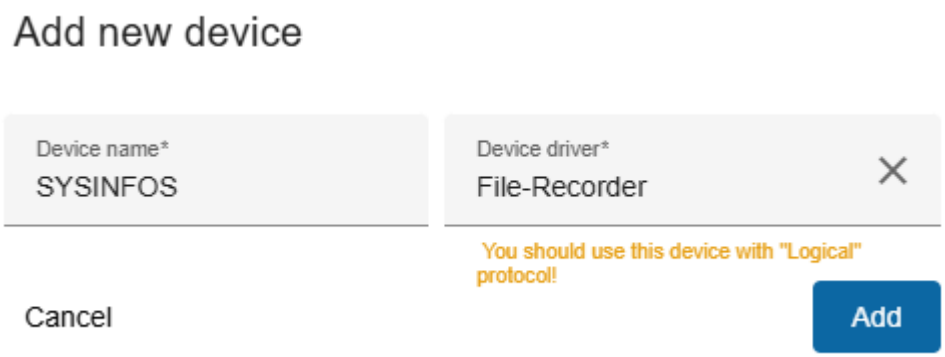
- **Trigger**:
  - when you click on the "Add interface +" button on the bottom to create new interface or
  - an on *pencil icon* to edit from context menu or
  - on *down arrow Add interface below*to add interface from context menu
  - The dialog title clearly indicates whether you are creating a new interface, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Port name**: The communication port / interface that thread should use.
    - **null**: The null interface is used by all logical devices. In every M&C system there is exactly one null interface running the Logical protocol. All logical devices are connected to this (physically not existent) interface.
  - **Idle time in msec**: The idle time of thread (msec).
  - **Protocol type**: The protocol for thread. It should be one of drop-down options.
- **Actions**:
  - **Add/Save**:
    - For new interface, this creates the interface with the provided details.

        - For existing interface, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

Add/edit device



- **Trigger**:
    - when you click on *Add device +* to add device from context menu or
    - an on *pencil icon* to edit from device from row or
    - The dialog title clearly indicates whether you are creating a new interface, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
    - **Device name**: The name of the device. It must be unique.
    - **Device driver**: The device driver. It should be one of the option from the drop-down list. If you select the wrong device driver for the interface protocol, a warning is displayed and it is displays which protocol you should use for this device like in image below.



- **Actions**:
    - **Add/Save**:

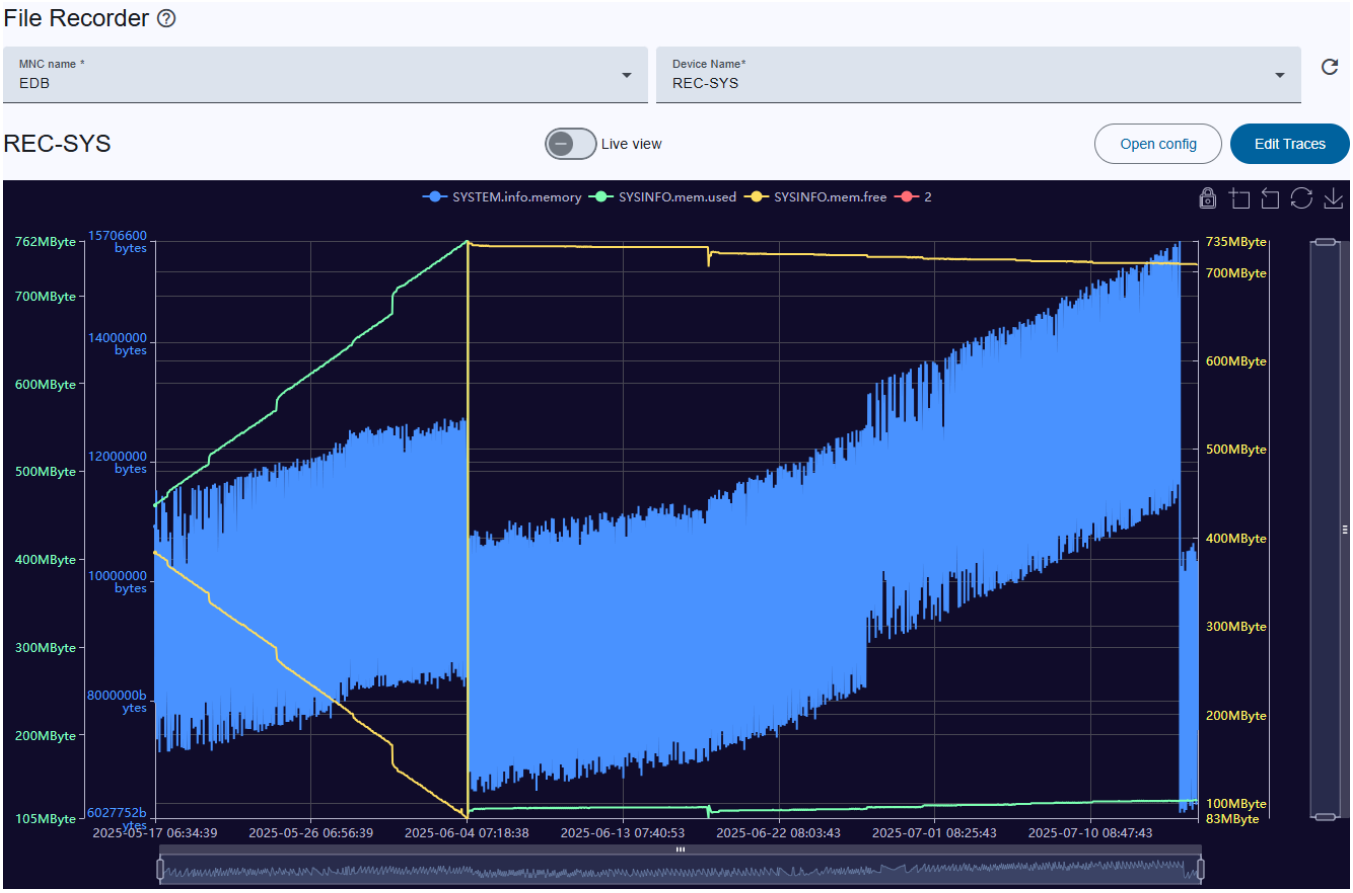
        - For new device, this creates the device with the provided details.
        - For existing device, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

Confirmation dialog

- **Trigger**: Opens when you click the
    - Trash icon for an device from menu column
    - Trash icon from bottom toolbar.
    - Reset button
    - Leaving the page if there are unsaved changes.
- **Content**: Asks the user to confirm the action, depending on where you trigger the dialog.
- **Actions**:
    - **Yes**: Confirm the action.
    - **Cancel**: Cancels the action.

2.8 File recorder

The sat-nms M&C software provides a facility (a logical device) called File-Recorder which is capable to record arbitrary data like levels, antenna pointing etc. into a disc file. The File recorder page is the user interface to inspect such files. The page can be accessed either from the side toolbar File-Recorder menu or can be configured in user-screens on button-element as File-Recorder action.



Top toolbar

- **MNC name**: A dropdown to select the available MNC from drop-down.
- **Device name**: A dropdown to select the available file-recorder devices from drop-down.
- **Reload**: A button to reload the file-record data.

Title, live view, and configuration buttons

- **Title**: Displays the currently selected device name (e.g., "REC-SYS").
- **Live view toggle**: Beside for viewing stored log files, the File recorder Viewer can be used to display the logged data as it is acquired with an automated display update. By pressing the "live view" toggle button, the viewer changes to live view mode. In this mode the viewer shows up to 400 recent data points, automatically updated with an interval that is configurable separately from the standard interval at the device window setup (open config button).
- **Configuration buttons**:
  - **Open config**: Opens a device window allowing users to view or modify settings.
  - **Edit traces**: Opens a dialog for customizing or editing trace parameters, such as specific trace name and their units.

Graph

- **Legends**:
  - Positioned at the top of the graph, legends display the names of the traces.
  - **Interactive Features**: Clicking on a legend hides or shows the corresponding trace line on the graph for better visualization.
- **Zoom actions**: *Described below under zoom control buttons (near the legend)*
- **Graph lines and axes**:
  - **Graph lines**: Display the traces values over time, represented in different colors as per the legend.
  - **Y-axis**:
    - **Left**: Displays the primary value ranges for the traces.
    - **Right**: Displays secondary value ranges for specific traces.
  - **X-axis (bottom)**: Represents the timeline of the data being monitored.
- **Zooming**
  - **Lock Y-Axis**: By clicking on the lock symbol, the y-axis is locked and the user can interact with the diagram without scaling.
  - **Scrolling in and out**: Use the scroll wheel to zoom in or out horizontally along the timeline (X-axis).
  - **Vertical zoom (y-axis right side)**: Adjusts the zoom level vertically, allowing focus on specific Y-axis value ranges.
  - **Horizontal zoom (x-axis bottom)**: Slider enables horizontal zooming along the timeline, allowing you to focus on specific time intervals.
  - **Zoom control buttons (near the legend)**:
    - **Zoom brush (left)**: Enables manual zooming by selecting an area on the graph.
    - **Undo zoom**: Resets the graph to the previous zoom level.
    - **Restore**: Resets the graph to its default view (original scale).
    - **Save as image**: Exports the current graph as an image file.

2.8.1 Functionality

Graph interaction

- **Zooming**
  - **Sliders**: A slider bar (bottom and right) is provided, on which coordinate systems can be zoomed or roamed by mouse dragging.
- **Scroll**: Scrolling inside zooms in on the graph and scrolling out zooms out on the graph.

Tooltip: Displaying data reference to line and axis value under mouse pointer.

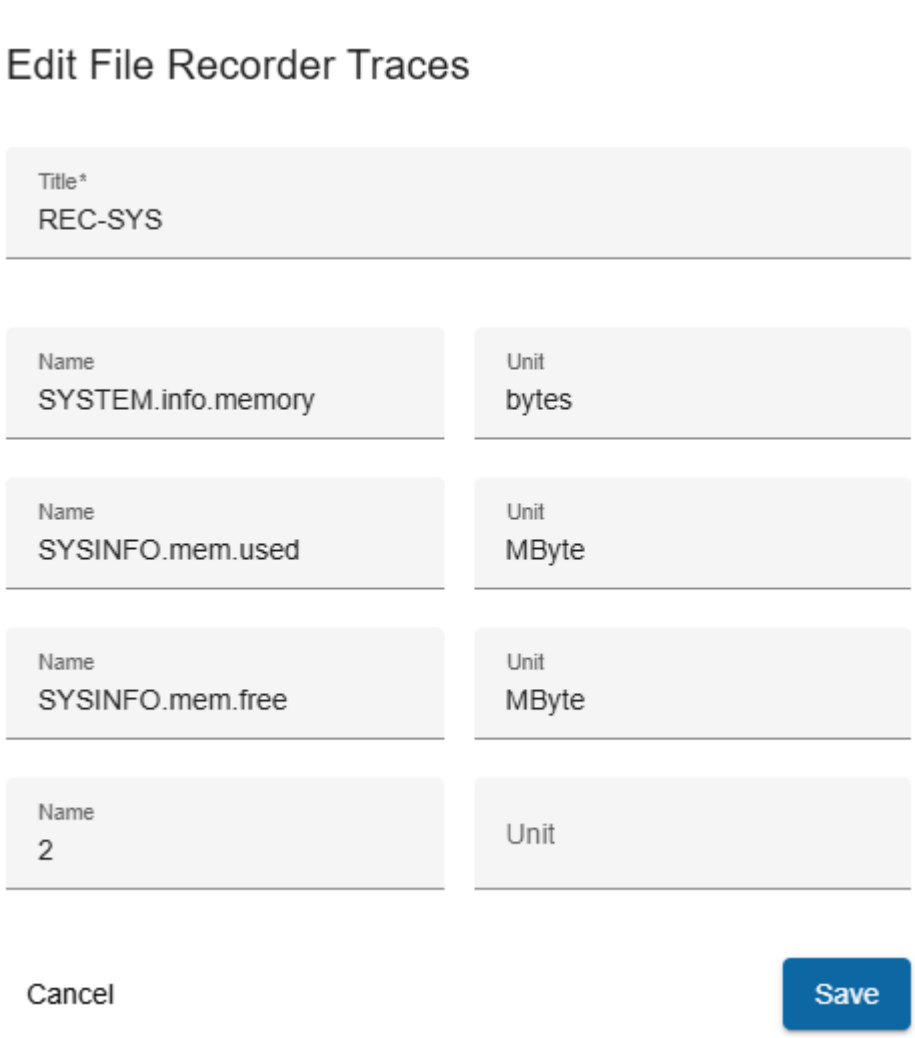
Legends: Clicking on a legend hides or shows the corresponding trace line on the graph for

better visualization.

2.8.2 Dialogs

Open config: Opens a device window config page.

Configure traces



- **Trigger**: when you click on the *Edit traces* button.
- **Fields**:
  - **Title**: The title of selected file-recorder device. *NOTE*: This is differ from device name.
  - **Name**: Free text to set trace name.
  - **Unit**: The unit of trace.
- **Actions**:
  - **Save**:
    - Saves modifications.
    - If an error occurs in the input fields, the button is deactivated.

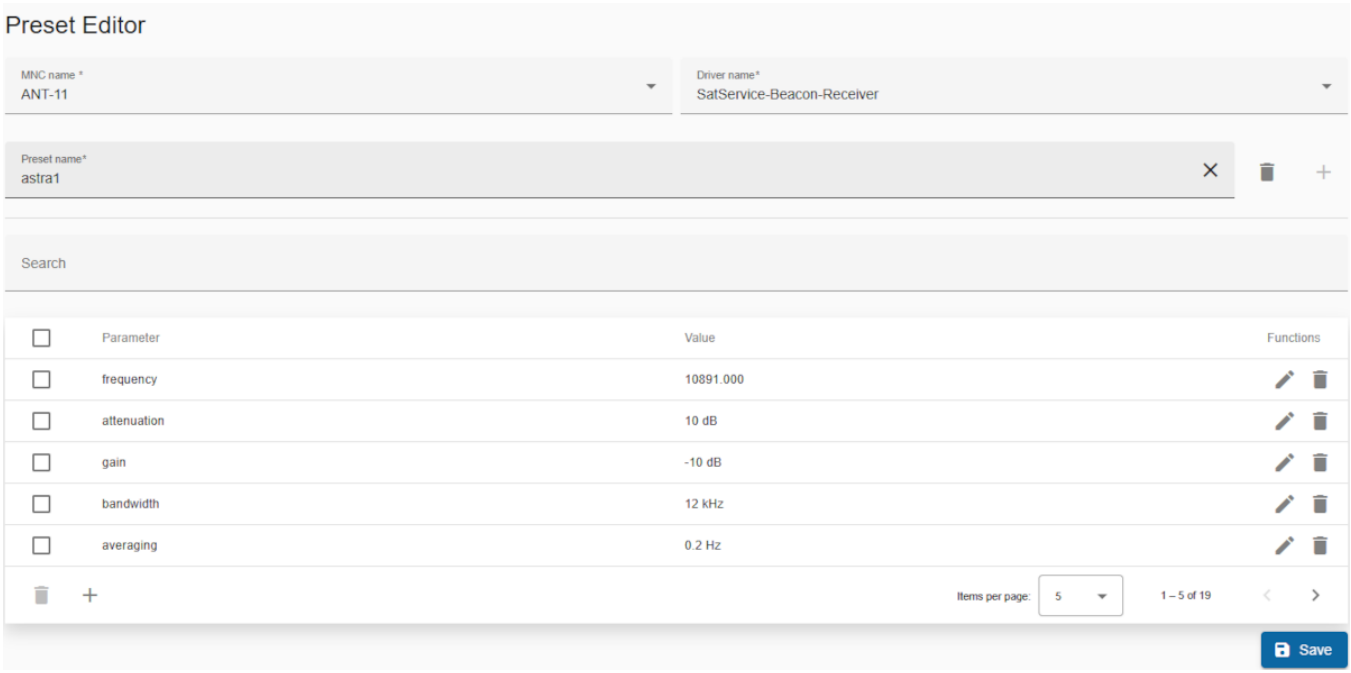
- **Cancel**: Closes the dialog without saving.

2.9 Preset editor

Most sat-nms device drivers allow you to save a snapshot of the device settings in*device preset files* so that the operator can restore these settings at a later time.

The presets are stored on the server that your client programme connects to. In an M&C environment this is the M&C server, in an NMS environment this is the central NMS server. This means that in the latter case, a preset can be saved when operating a VLC and can later be applied to a device on another VLC.

Presets are always assigned to the device type (driver) with which they were recorded. This is because it makes no sense to transfer the settings of a beacon receiver to a DVB encoder. The target device does not *know* what to do with the receiver settings. Even with similar devices (e.g. two IRDs from different manufacturers), there may be differences in the number and type of settings that the devices understand. The software therefore saves the preset files in a structure *by device driver*.



Title: Title of this page.

MNC name: Below the title, you must select an option from the drop-down list.

Driver name: A dropdown with all driver names. Just select from dropdown.

Preset name: Also a dropdown list of preset file names with the search function.

- **Delete**: A *trash can icon* can be used to delete selected preset file.
- **Add preset**: A + button can be used to add Prest. To activate the + button when it is deactivated, simply delete the input of the preset name.

To load a stored preset first you have to select MNC name, then driver name and then preset name. After selection, a table with the saved presets would de displayed, as shown in the image above.

Search bar

- Located at the top of the table.
- Allows users to filter events by keywords.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
    - **Location**: The checkbox column is located at the far left of the table.
    - **Purpose**: Allows users to select one or multiple rows for delete row(s).
    - **Functionality**:
        - **Single Selection**: Click on a single checkbox to select an individual row.
        - **Multiple Selection**: Click multiple checkboxes to select multiple rows at once.
        - **Select all**: The checkbox in the header row allows you to select or deselect all rows on all pages.
        - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **Parameter**: Stored parameter for the preset.
- **Value**: Stored value of parameter.
- **Functions**: Action icons for managing the parameter:
    - **Edit**: Pencil icon to modify/edit the parameter.
    - **Delete**: Trash can icon to delete the parameter.

Bottom toolbar

- **Delete selected rows**: A button to delete selected rows from checkbox column.
- **Create**: + button is used to create new parameter.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

Save: Saves all modifications. *NOTE: You must save the changes before leaving the editor, otherwise the changes will be lost.*

2.9.1 Functionality

Create preset

- Click the + button near the delete icon and preset name input to add a new preset.
- Input the required details.

Create parameter

- Click the + button at the bottom of the table to add a new parameter.
- Input the required details.

Edit parameter

- Click the *pencil icon* in the Functions column to modify parameter details.
- Adjust the settings and save.

Delete preset

- Click the *trash icon* near the preset name input to delete the preset.
- Confirm the action in the pop-up dialog to delete the parameter.

Delete parameter

- Click the *trash icon* in the Functions column to delete an parameter.
- Confirm the action in the pop-up dialog to delete the parameter.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.9.2 Dialogs

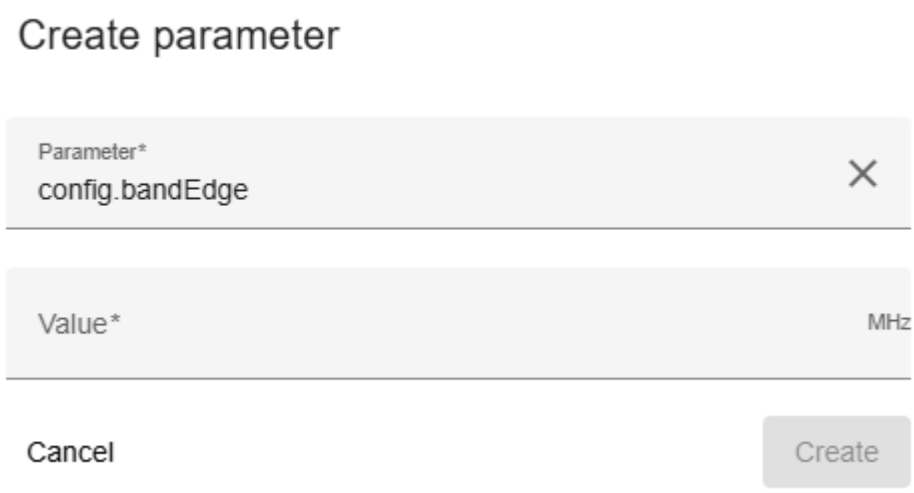
Add preset dialog

- **Trigger**: When you click on the "+" near the delete icon and preset name input to add a

new preset create new preset.

- **Fields**:
  - **Name**: The user defined name of preset. Name must be unique compared to other presets.
- **Actions**:
  - **Add**: Adds the new preset.
  - **Cancel**: Closes the dialog without saving.

Add/edit parameter



- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new parameter or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new parameter, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Parameter**: Parameter should be selected from dropdown list. As soon as you have selected a parameter value, the value input will be displayed.
  - **Value**: The parameter value van be added.
- **Actions**:
  - **Create/Save**:
    - For new parameter, this creates the parameter with the provided details.
    - For existing parameter, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

Confirmation dialog

- **Trigger**: Opens when you click the
  - delete preset
  - delete parameter
- **Content**: Asks the user to confirm the action, depending on where you trigger the dialog.
- **Actions**:
  - **Yes**: Confirm the action.
  - **Cancel**: Cancels the action.

2.10 Inventory database

The inventory database provides a structured view of the drivers, so that the user can easily track their state and documents that upload by the user.

The title bar provides access to the two different datasets:

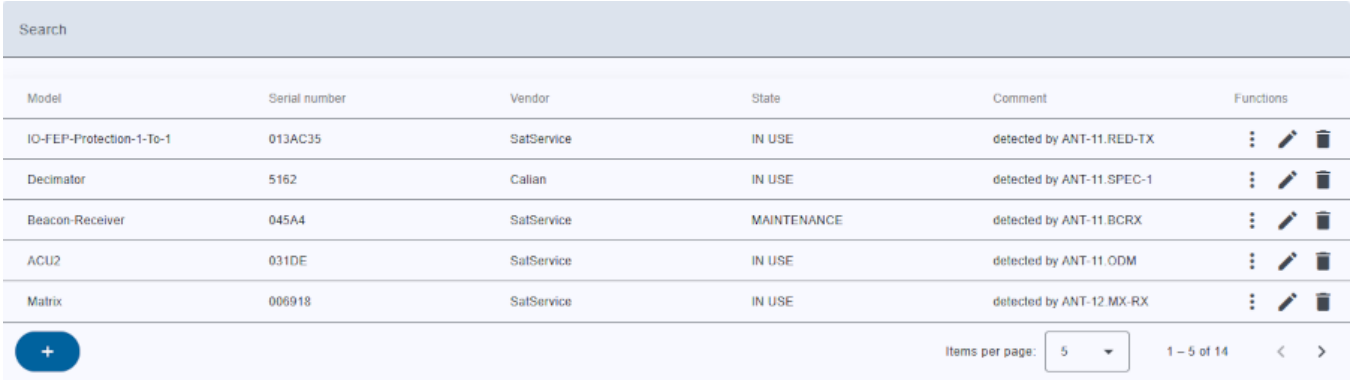
- Inventory (default)
  - displays the list of inventory drivers
- Documents
  - upload document
  - list of uploaded documents

You can switch between these different tables by clicking on the buttons at the top of this page.

On a system without installed SatDB or if the PostgreSQL Database is not reachable an error message will be display: Error DB not reachable. In this case, please check your database installation.

2.10.1 Inventory

The inventory table provides a structured view of the drivers automatically found by M&C or added by the user, so that the user can easily track their state.



To enable the Inventory configure a following in the filevlc.properties with:

- *db.inventory.enabled=true*
- *db.inventory.debug=false*

When the inventory is activated, the M&C automatically recognizes the drivers for whom the serial number has been recognized by M&C and automatically adds them to the database. Non-Recognized driver should be added by the user.

Inventory items are stored by there vendor and model name. For this reason, the item should only exist once.

The Inventory list also contains devices and logs.

- Driver recognized by M&C also check for the devices that are configured on the driver and add to database. If not user can also add manually.
- Logs are also added automatically if M&C has recognized the driver. However, the user can also add the log entry manually. For example, when adding a device that M&C has not recognized automatically. You can see device(s) and logs by clicking on the row of the inventory table.

Search bar

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all list and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

The inventory table contains the following columns:

Column name	Description
Model	name of model
Serial number	Serial number of driver
Vendor	name of vendor
State	State of driver
Comment	Free text describing the situation

The last column located at the far right of the tables provides buttons to run actions on the

corresponding inventory item.

The following functions for inventory are `available` :

Function	Icon	Description
Drop down		Go directs to documents page with selected action.
Edit		modify/edit inventory item
Delete		to delete the inventory item. A confirmation dialog will appear to confirms the action.

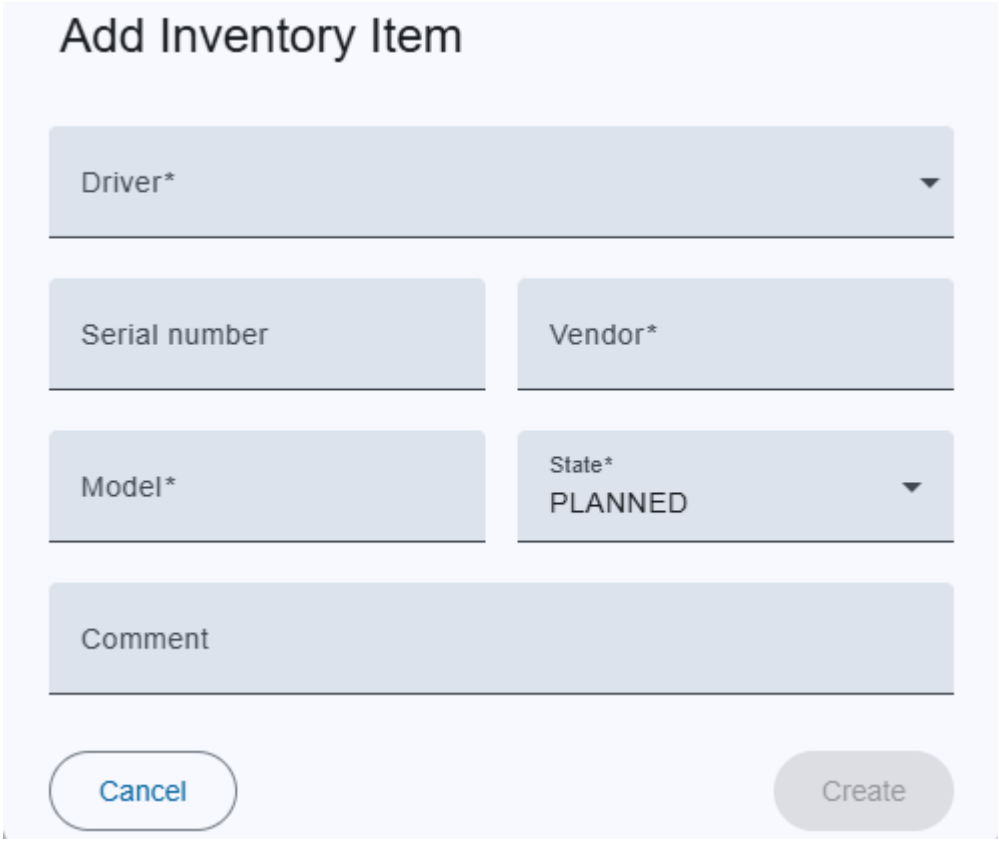
Clicking on a inventory item row shows other relatives tables:

- Devices
- Logs

Bottom toolbar

- **Create**: + button is used to create new inventory item.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

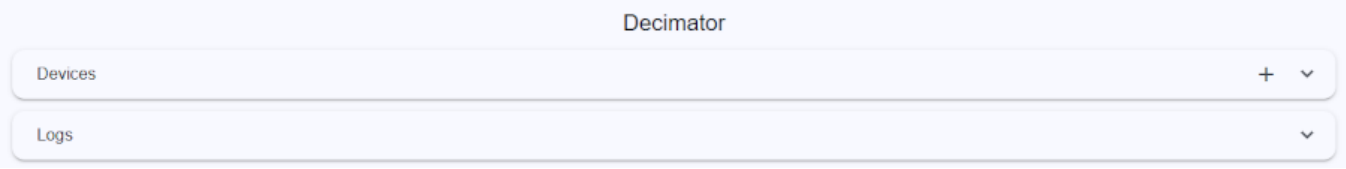
2.10.1.1 Add/edit inventory dialog



- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new inventory item or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new inventory item, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Driver**: An drop down with available drivers. Already exists driver in inventory list option is disabled.
  - **Serial number**: Serial number of driver
  - **Vendor**: Manufacture name. Automatically filled if you select an driver but it can also changes.
  - **Model**: Model is driver name.Automatically filled if you select an driver but it can also changes.
  - **State**: State of the driver.
  - **Comment**: Free text for providing some information.
- **Actions**:
  - **Create/Save**:
    - For new inventory item, this creates the inventory item with the provided details.

        - For existing inventory item, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

2.10.2 Inventory related tables



Toolbar (top)

- **Title**:
    - Model name
- **Tables**:
    - Devices
    - Logs
    - **Functionality**: Clicking on title or + or arrow to downside expands the table.

2.10.2.1 Devices table

Devices table show the list of devices that are configured on the selected inventory driver. Driver recognized by M&C check for the devices that are configured, add to database. By non-recognized driver devices should be added manually.

Top

- Left-side title of table.
- Plus (+) button to create new device.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon

near the column name.

Table columns

The devices table contains the following columns:

Column name	Description
MNC	Name of M&C in which the device is configured.
Device name	Device name
Online	State of device. If device have an communication fault state will be false else true.

The last column located at the far right of the tables provides buttons to run actions on the corresponding device.

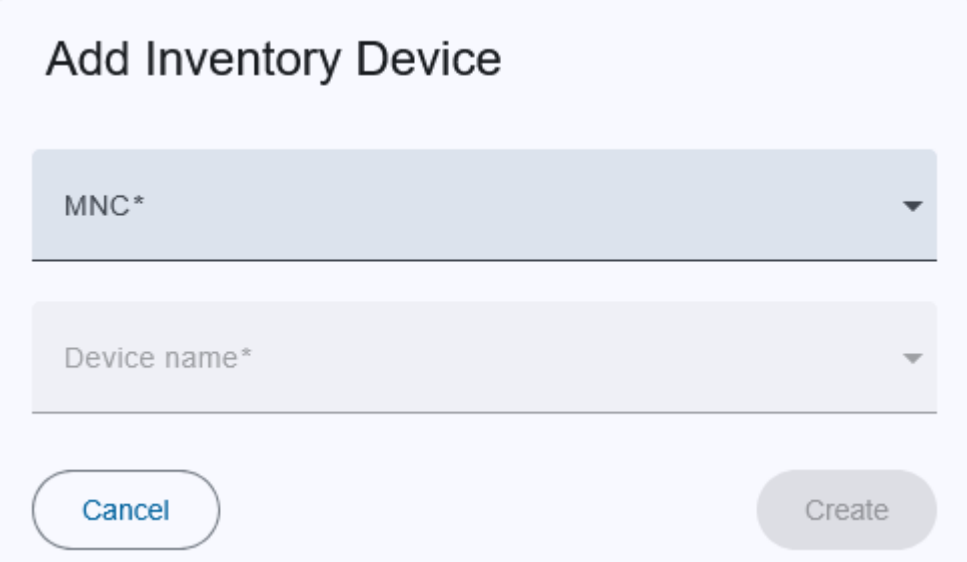
The following functions for device are `available` :

Function	Icon	Description
Edit		modify/edit device
Add log		Opens Add Inventory Log dialog with prefilled data to add a log in inventory database.
Delete		to delete the device. A confirmation dialog will appear to confirms the action.

Bottom toolbar

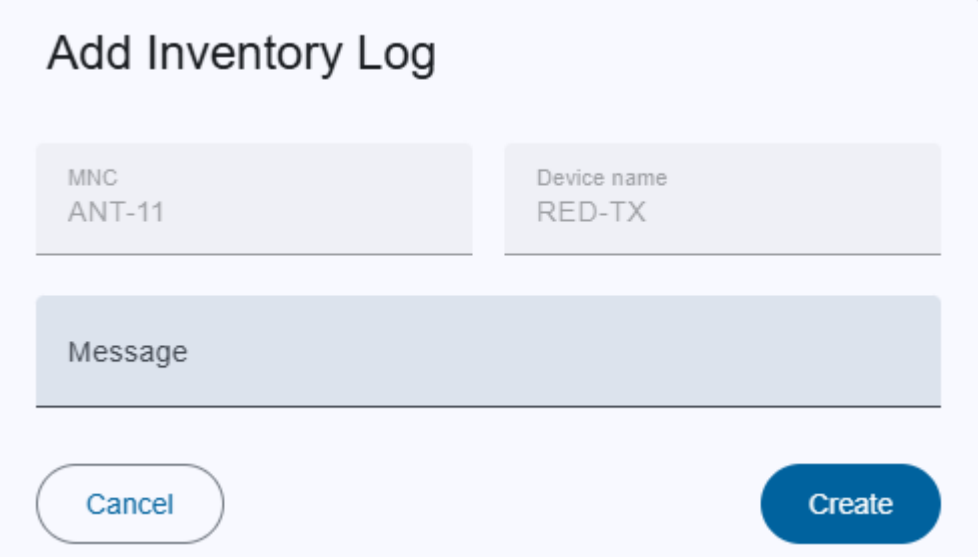
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.10.2.2 Add/edit device dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit device in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **MNC**: Name of the M&C.
  - **Device name**: Once you selected the M&C device list will be available on behalf of selected M&C and selected Driver of inventory item.
- **Actions**:
  - **Create/Save**:
    - For new device, this creates the device with the provided details.
    - For existing device, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

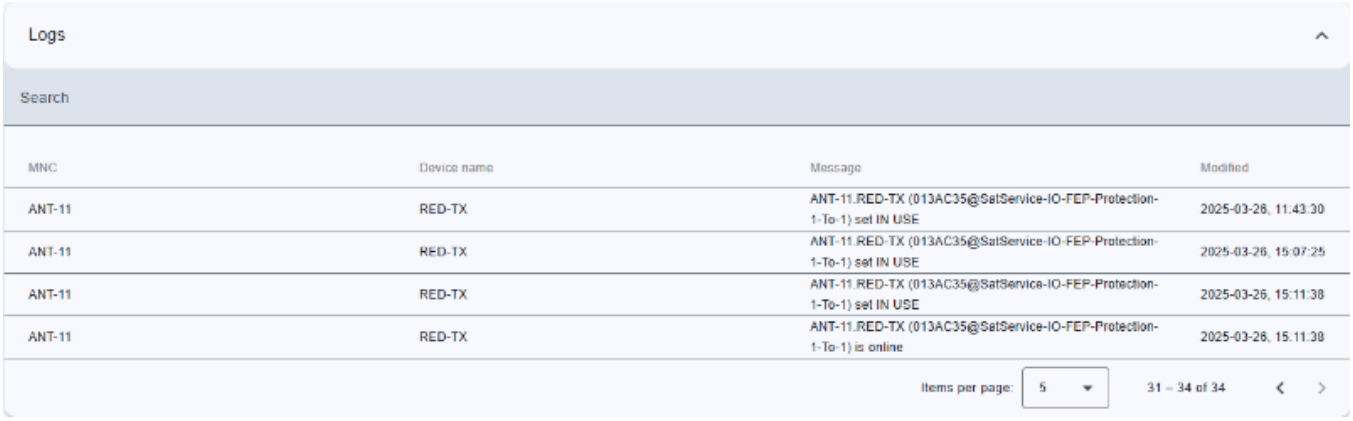
2.10.2.3 Add log dialog



- **Trigger**:
  - when you click on the button on the device column
- **Fields**:
  - **MNC**: Name of the M&C.
  - **Device name**: Device name.
  - **Message**: A short message that shows the purpose of this log.
- **Actions**:
  - **Create**:
    - this created a new log.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

2.10.2.4 Logs table

The logs helps to trace the activities of device(s) and driver. Logs are also added automatically if M&C has recognized the driver. However, the user can also add the log entry manually. For example, when adding a device that M&C has not recognized automatically.



Top

- Left-side title of table.
- Icon button to expand or collapse table

Search bar

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all list and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

The Logs table contains the following columns:

Column name	Description
MNC	Name of M&C.
Device name	Device name.
Message	Purpose of this log.
Modified	Date of added log.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.11 Documents

In Inventory database you can also upload documents and download documents. Documents can be attached to M&C, Driver or Globally.

Expand All: Expand all category tables.

Import file: Import file trigger *Upload document* dialog.

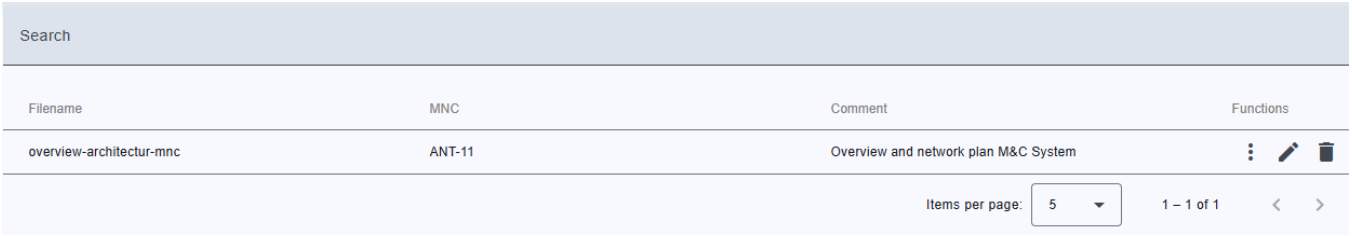
Global : If the document is uploaded without M&C or driver name, it belongs to Global category.

MNC : If the document is uploaded with, it belongs to MNC category.

Driver : If the document is uploaded with driver name, it belongs to Driver category.

Tables

The tables for Global, MNC and Driver are identical.



Search bars

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all list and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

The document table contains the following columns:

Column name	Description
Filename	Name of M&C.
MNC/Driver	Depending which section you expanded MNC/Driver will shown.
Comment	Purpose of document.

The last column located at the far right of the tables provides buttons to run actions on the corresponding inventory item.

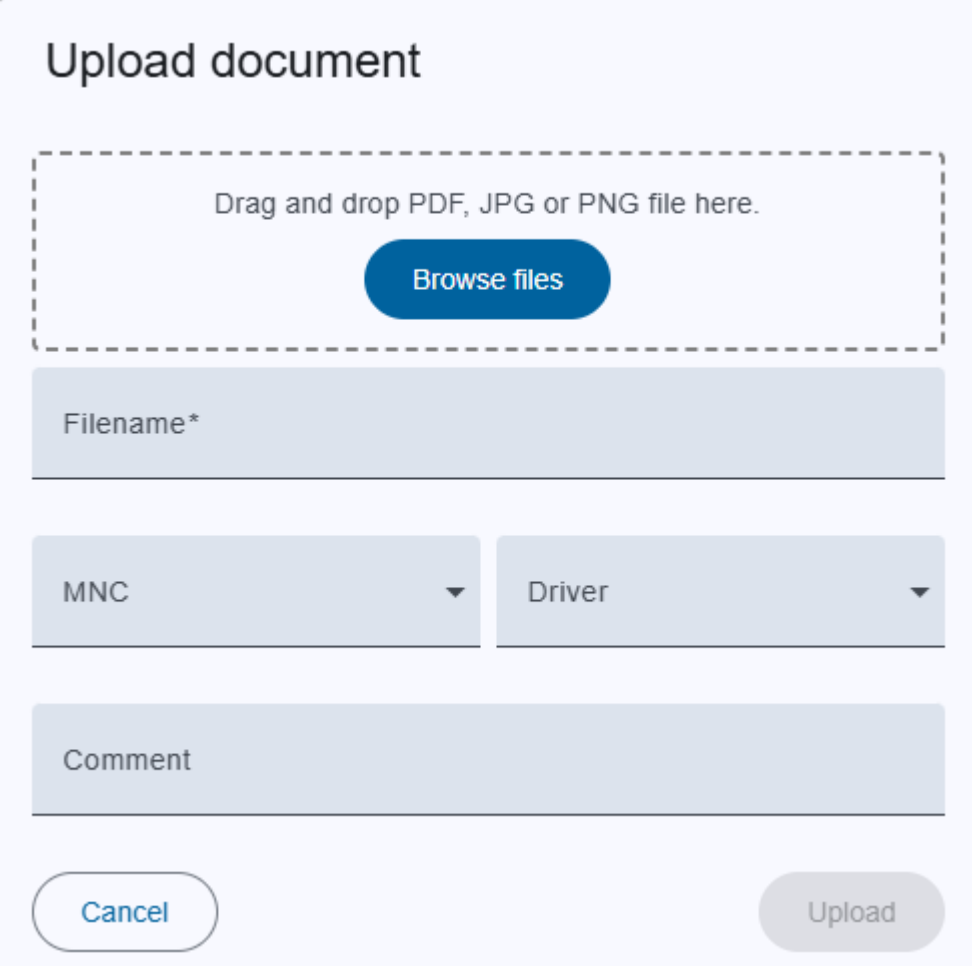
The following functions for inventory are `available` :

Function	Icon	Description
Drop down		Shows the option to download document or open document in new tab.
Edit		modify/edit document item
Delete		to delete the document. A confirmation dialog will appear to confirms the action.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.11.1 Upload/edit document



- **Trigger**:
  - when you click on the "upload" button on the top toolbar to upload or
  - from drop down ⋮ "Upload file" or
  - an on *pencil icon* to edit document in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **File**: You can just drop the file in section or just browse the files by clicking on "Browse files". Supported format are *pdf, png, jpg*.
  - **Filename**: Once you selected the file or dropped the file, filename will be taken from dropped file. Can also be changed.
  - **MNC**: M&C name to whom document belongs. If driver name is selected M&C input will be disabled.
  - **Driver**: Driver name to whom document belongs. If M&C name is selected M&driver input will be disabled.
  - **Comment**: Some information about document.

NOTE: If M&C and Driver fields are empty, the document will belongs to*Global Category*.

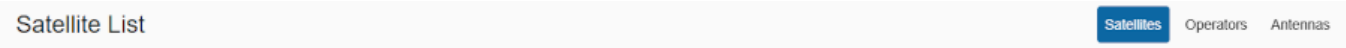
Edit document: If you want to change the document properties but do not want to upload the document, you do not need to drop or browse the "File" section.

- **Actions**:
  - **Upload**:
    - this upload document.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

2.12 Satellite database

The Satellite database provides a structured view of satellites, operators, and antennas. Each category is represented by a distinct table. Users can navigate between these tables using the menu options and access additional details for a selected satellite.

The title bar provides access to the three different datasets:



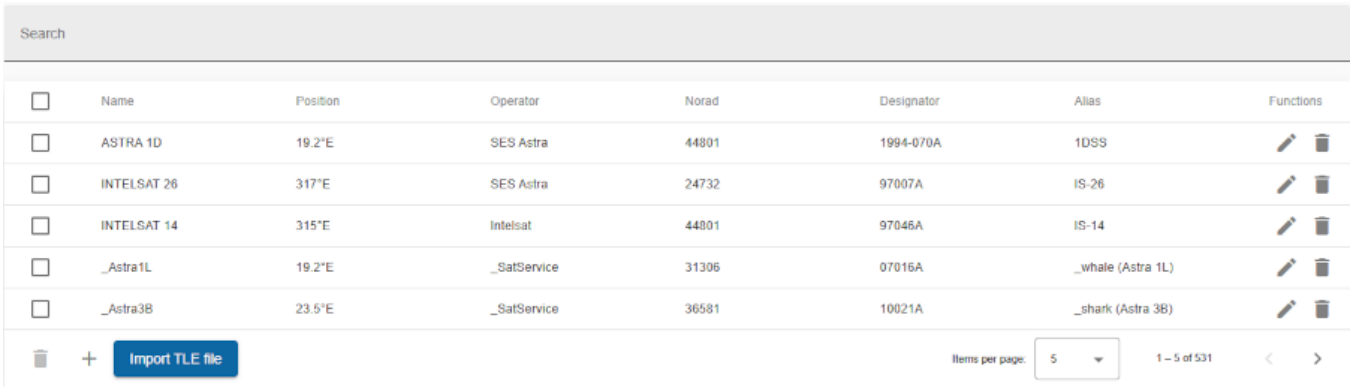
- Satellites (default)
  - displays the list of satellites stored in the database
- Satellite operators
  - each satellite belongs to an satellite operator
  - here you can store for example contact information for line-ups
- Antennas
  - list of configured antennas (and the related devices) which should use the database
  - here you defined the satellite positions an antenna can use to point to a satellite

You can switch between these different tables by clicking on the buttons at the top of this page.

On a system without installed SatDB or if the PostgreSQL Database is not reachable an error message will be display: Error DB not reachable. In this case, please check your database installation.

2.12.1 Satellite list

This shows the list of all satellites in the database.



Satellites are stored by there unique name. We recommend to use the names used by CelesTrak, because this allowes easy bulk import of TLE (two-Line elements). If you prefere deferent name, use the optional Alias field.

The satellite list itself does not contain any satellite position because they depends on the individual antenna. These positions can be found on the Antenna list page.

To use this satellites to move antennas you have to configure the corresponding Antenna Management Devices on the MNC server(s) which controlls the ACU (antenna control unit):

- Antenna-Management

Search bars

- Located at the top of the table.
- Allows users to filter by:
  - *Keywords*: full-text search over all channels and all columns

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

The first column located at the far left of the table provides a checkbox to select one or more channels (rows) and perform the same operation on all selected channels.

- **Single Selection**: Click on a single checkbox to select an individual satellite
- **Multiple Selection**: Click multiple checkboxes to select multiple satellite.
- **Select all**: The checkbox in the header row allows you to select or deselect all satellites on all pages.

The following function(s) are available:

Function	Icon	Description

Function	Icon	Description
delete		delete the selected satellite from the database

The last column located at the far right of the tables provides buttons to run actions on the corresponding satellite.

The following functions for managing a satellite are `available` :

Function	Icon	Description
Delete		to delete the satellite
Edit		modify/edit satellite

The following buttons for global function(s) are available at the the bottom of the table:

Function	Icon	Description
Add		add a new satellite
Delete		to delete all selected the satellite
Import	Import TLE file	[import TLE] data from file
Page	Pagination controls	set the number of items displayed per page (e.g., 5, 10, or 25) and navigate between pages

The satellite table contains the following columns:

Column name	Description
Name	name of satellite
Position	satellite orbit position in degrees
Operator	name of satellite operator
Norad number	NORAD Catalog Number
Designator	International Designator
Alias	Customized alias name

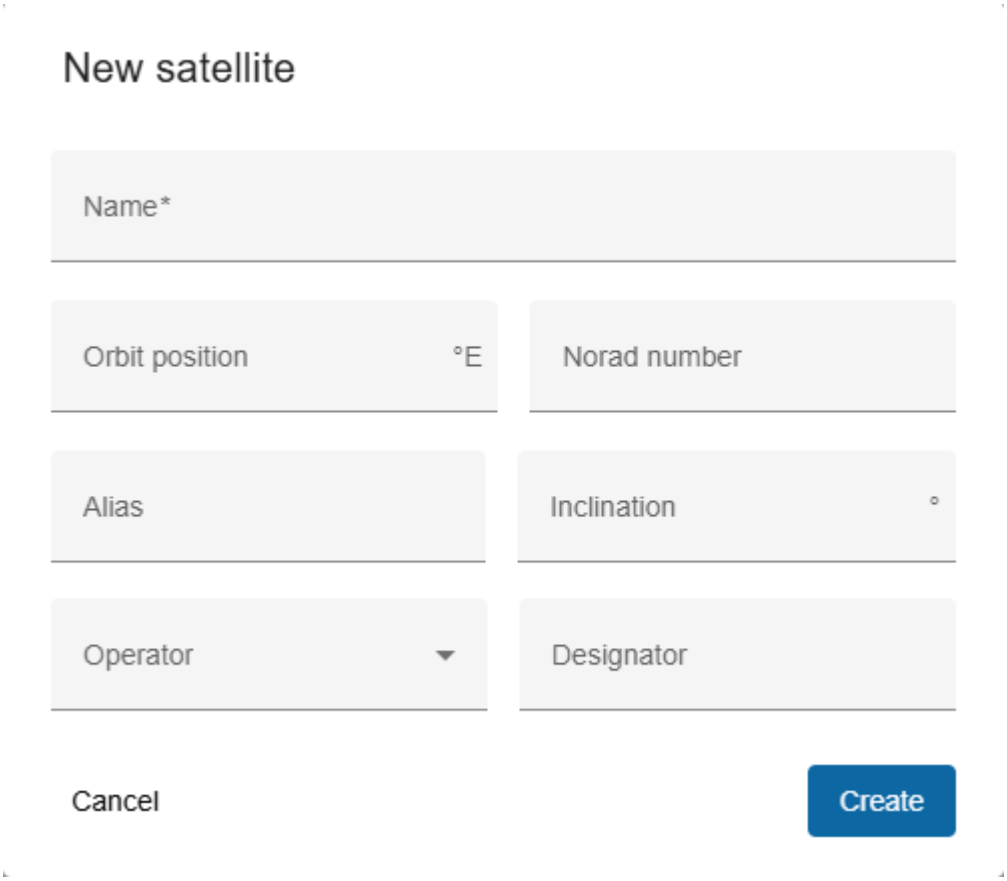
Clicking on a satellite row shows other relatives tables:

- Satellite position
- I11
- Beacons
- TC channels
- TM channels of selected row

Bottom toolbar

- **Create**: + button is used to create new satellite.
- **Import TLE file**: Opens a dialog to import TLE parameters.

2.12.2 Add/edit satellite dialog



- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new satellite or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new satellite, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Name**: The user defined name of this satellite.
  - **Orbit position**: Satellite orbit position in degrees.

- **Norad number**: unique name (world wide standardized)
- **Alias**: This can be used by customer to add own name.
- **Inclination**: Satellite inclination in degrees.
- **Operator**: Add satellite operator.
- **Designator**: Unique name (world wide standardized).

- **Actions**:
  - **Create/Save**:
    - For new satellite, this creates the satellite with the provided details.
    - For existing satellite, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

2.12.2.1 On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column or trash icon from bottom toolbar.
- **Content**: Asks the user to confirm whether they want to delete the selected satellite(s).
- **Actions**:
  - **Yes**: Deletes the satellite(s) permanently.
  - **Cancel**: Cancels the deletion action.

2.12.3 Satellite related tables

Toolbar (top)

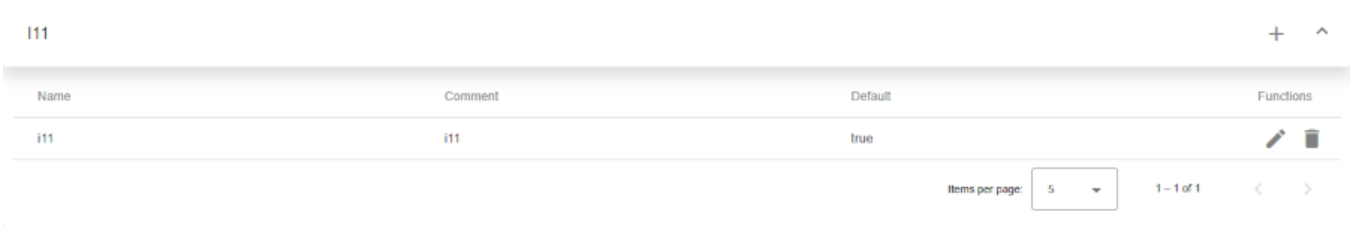
- **Title**:
  - Satellite name
  - Satellite alias
- **Edit NORAD TLE**: Trigger dialog to modify NORAD TLE.

Tables:

- I11 table
- Beacons
- TC channels
- TM channels
- **Functionality**: Clicking on title or + or arrow to downside expands the table.

2.12.3.1 I11 table

Intelsat 11 elements describing the satellite path for a given time. It is comparable to TLE (Two-Line Elements) but it is possible to have multiple I11 elements per satellite for the same time but with different accuracy. It may be useful to store very accurate I11 elements but they are only valid for a short time and in case that you are not able to update regularly, you can store a long term I11 in the database to be prepared if the short term element is expired.



Top

- Left-side title of table.
- Plus (+) button to create new i11.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Name**: The name of the satellite as stated in the original I11 parameter set.
- **Default**: True means this is the default I11 dataset for this satellite.
- **Comment**: Describing the purpose of this dataset.
- **Functions**: Action icons for managing the key:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.12.3.1.1 Functionality

Create i11

- Click the + button at the top of the table to add a new i11.
- Input the required details.

Edit i11

- Click the *pencil icon* in the Functions column to modify i11 details.
- Adjust the settings and save.

Delete i11

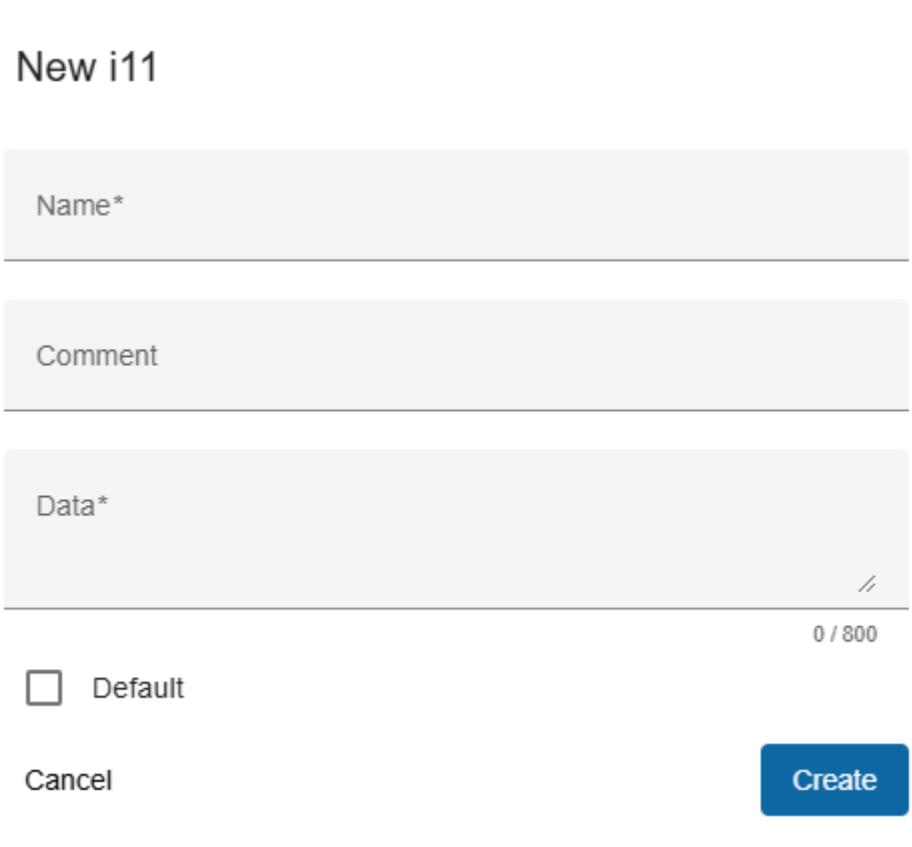
- Click the *trash icon* in the Functions column to delete an i11.
- Confirm the action in the pop-up dialog to delete the i11.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.12.3.1.2 Dialogs

Add/edit i11 dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit i11 in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:

- **Name**: The name of the satellite as stated in the original I11 parameter set.
- **Comment**: Free text describing the purpose of this dataset.
- **Norad number**: unique name (world wide standardized)
- **Data**: The 11 ephemeris parameters, floating point, separated by semicolon characters.
- **Default**: Set i11 as default.

- **Actions**:

  - **Create/Save**:
    - For new i11, this creates the satellite with the provided details.
    - For existing i11, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected i11.
- **Actions**:
  - **Yes**: Deletes the i11 permanently.
  - **Cancel**: Cancels the deletion action.

2.12.3.2 Beacons table

Satellite beacons are used to track satellites. Each satellite can have more than one beacon, usually there are at least one beacons on every polarization. The preferred beacon can be marked as default in the database. It will be used if the operator does not select a specific beacon for Step-Track.

Top

- Left-side title of table.
- Plus (+) button to create new beacon.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Frequency**: Beacon frequency, floating point, MHz.
- **Polarization**: This is an enumeration.
- **Comment**: Describing the purpose of this beacon.
- **Default**: True means this is the default beacon dataset for this satellite. .**Attenuation**: Modify attenuation value stored in the database for one particular pair of antenna and beacon. Opens beacon-attenuation dialog.
- **Functions**: Action icons for managing the key:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.12.3.2.1 Functionality

Create beacon

- Click the + button at the top of the table to add a new beacon.
- Input the required details.

Edit beacon

- Click the *pencil icon* in the Functions column to modify beacon details.
- Adjust the settings and save.

Attenuation

- Click *pencil outlined* in the row under attenuation. Modify value stored in the database for one particular pair of antenna and beacon. Opens beacon-attenuation dialog.

Delete beacon

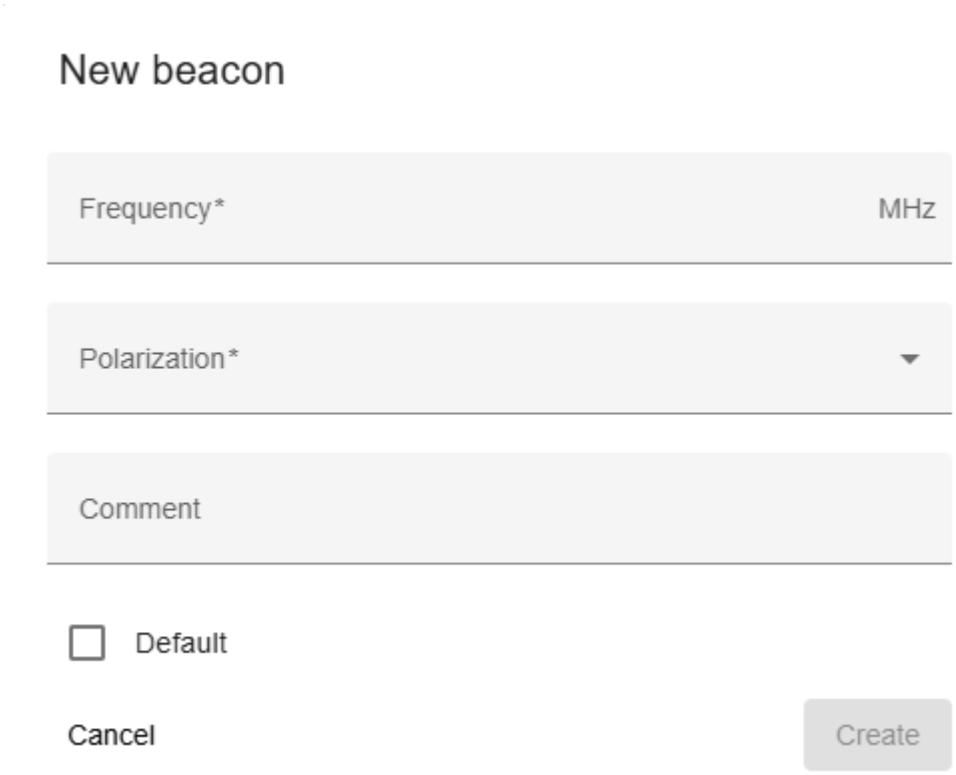
- Click the *trash icon* in the Functions column to delete an beacon.
- Confirm the action in the pop-up dialog to delete the beacon.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.12.3.2.2 Dialogs

Add/edit beacon dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit beacon in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Frequency**: Beacon frequency, floating point, MHz.
  - **Polarization**: An enumeration, one of "H", "V", "RHCP" or "LHCP".
  - **Comment**: Free text describing the purpose of this dataset.
  - **Default**: Set beacon as default.
- **Actions**:
  - **Create/Save**:
    - For new beacon, this creates the beacon with the provided details.
    - For existing beacon, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

Modify beacon attenuation

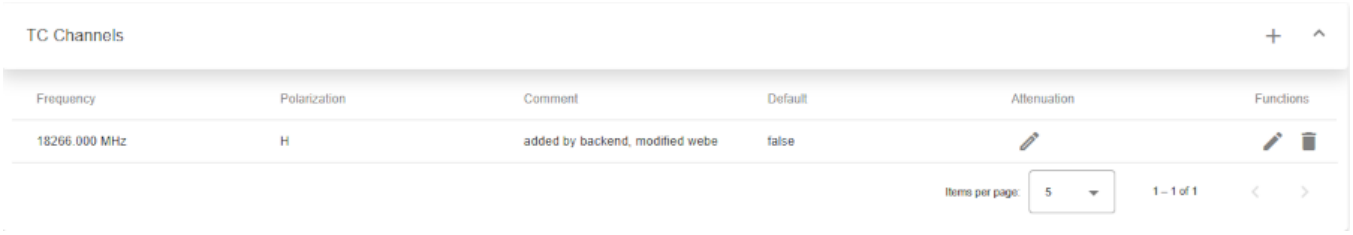
- **Trigger**: Click on the *pencil outlined* button in the row under *attenuation*
- **Fields**:
  - **Select antenna**: Drop-down to select antenna.
  - **Attenuation**: Receiver attenuation, floating point, dB.
- **Actions**:
  - **Save**:
    - Saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected beacon.
- **Actions**:
  - **Yes**: Deletes the beacon permanently.
  - **Cancel**: Cancels the deletion action.

2.12.3.3 TC channel table

Tele-Command (TC) channels defining communcation channels used to send commands to a satellite.



Top

- Left-side title of table.
- Plus (+) button to create new TC channel.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Frequency**: TC channel frequency, floating point, MHz.
- **Polarization**: This is an enumeration.
- **Comment**: Describing the purpose of this beacon.
- **Default**: True means this is the default beacon dataset for this satellite. .**Attenuation**: Modify attenuation value stored in the database for one particular pair of antenna and TC channel. Opens TC-Attenuation dialog.
- **Functions**: Action icons for managing the key:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.12.3.3.1 Functionality

Create TC channel

- Click the + button at the top of the table to add a new TC channel.
- Input the required details.

Edit TC channel

- Click the *pencil icon* in the Functions column to modify TC channel details.
- Adjust the settings and save.

Attenuation

- Click *pencil outlined* in the row under attenuation. Modify value stored in the database for one particular pair of antenna and TC channel. Opens beacon-attenuation dialog.

Delete TC channel

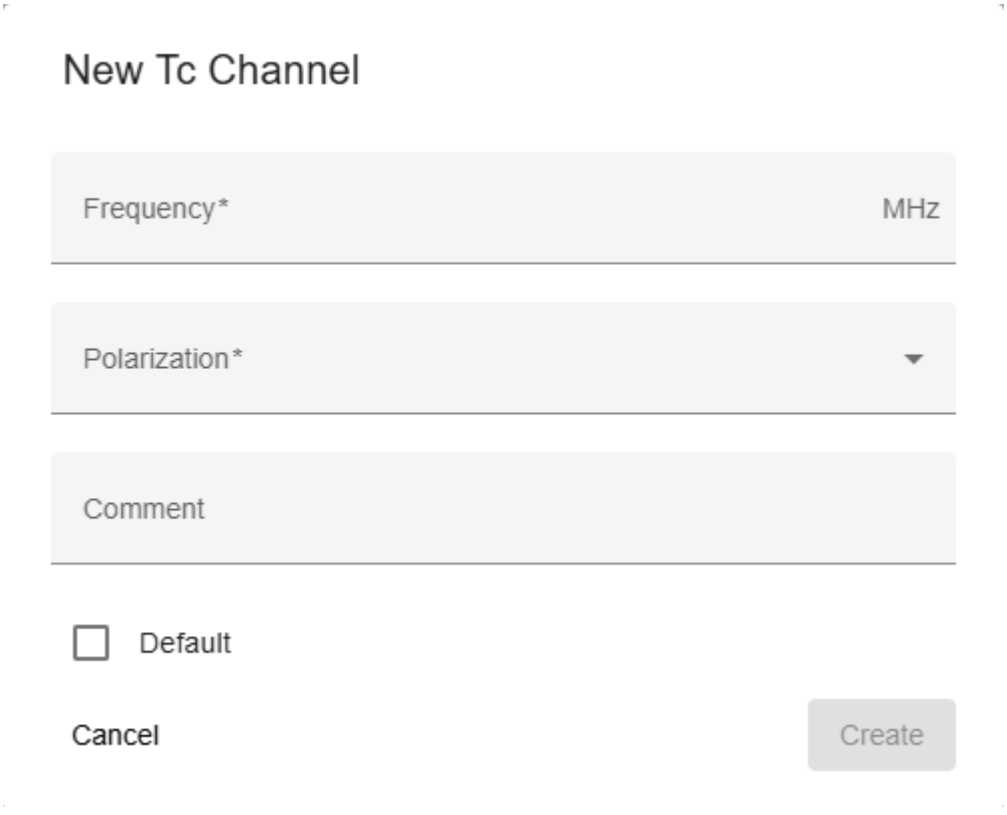
- Click the *trash icon* in the Functions column to delete an TC channel.
- Confirm the action in the pop-up dialog to delete the TC channel.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.12.3.3.2 Dialogs

Add/edit TC channel dialog

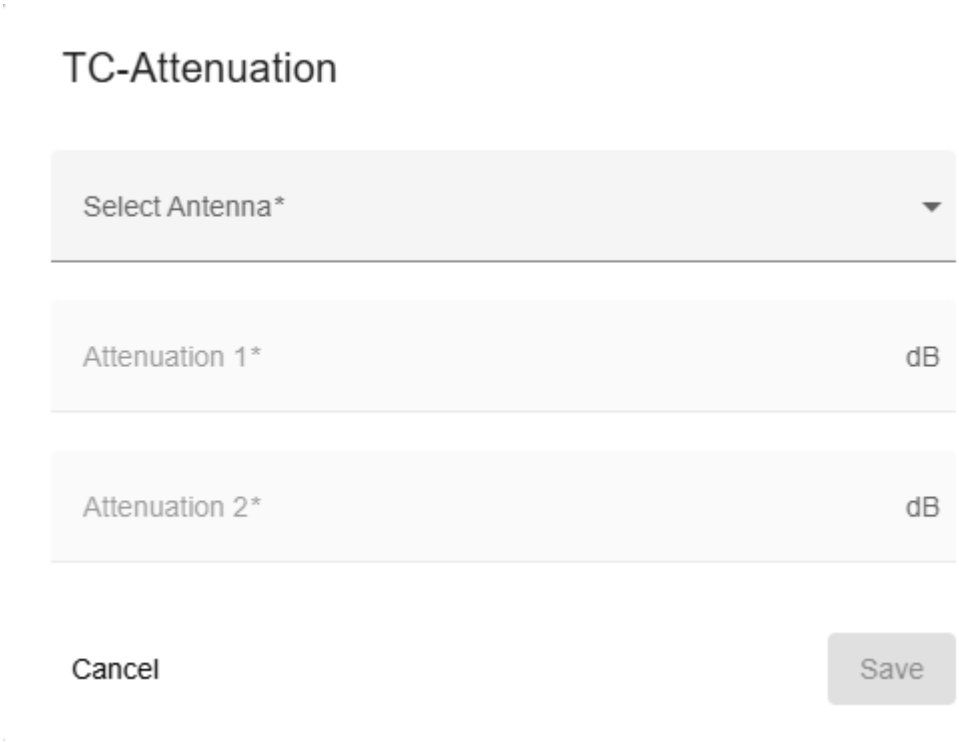


- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit TC channel in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:

- **Frequency**: TC channel frequency, floating point, MHz.
- **Polarization**: An enumeration, one of "H", "V", "RHCP" or "LHCP".
- **Comment**: Free text describing the purpose of this dataset.
- **Default**: Set TC channel as default.

- **Actions**:
  - **Create/Save**:
    - For new TC channel, this creates the TC channel with the provided details.
    - For existing TC channel, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

Modify TC channel attenuation



- **Trigger**: Click on the *pencil outlined* button in the row under *attenuation*
- **Fields**:
  - **Select antenna**: Drop-down to select antenna.
  - **Attenuation 1**: Transmit attenuation1, floating point, dB.
  - **Attenuation 2**: Transmit attenuation2, floating point, dB.
- **Actions**:
  - **Save**:
    - Saves modifications.
    - If an error occurs in the input fields, the button is deactivated.

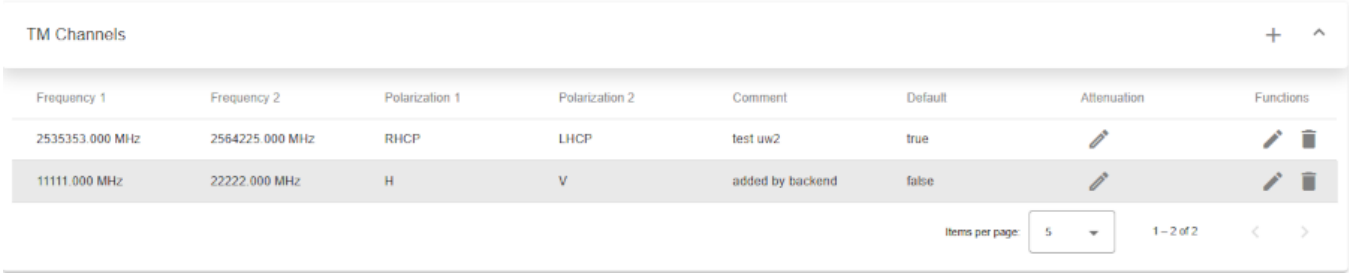
- **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected TC channel.
- **Actions**:
  - **Yes**: Deletes the TC channel permanently.
  - **Cancel**: Cancels the deletion action.

2.12.3.4 TM channel table

Telemetry (TM) channels defining communcation channels used to receive status and measurement data from satellites.



Top

- Left-side title of table.
- Plus (+) button to create new TM channel.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Frequency 1**: Primary receive frequency, floating point, MHz.
- **Frequency 2**: Secondary receive frequency, floating point, MHz.
- **Polarization 1**: This is an enumeration.
- **Polarization 2**: This is an enumeration.
- **Comment**: Describing the purpose of this beacon.
- **Default**: True means this is the default TM channel dataset for this satellite. .
  Attenuation: Modify attenuation value stored in the database for one particular pair of antenna and TM channel. Opens TC-Attenuation dialog.
- **Functions**: Action icons for managing the key:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.12.3.4.1 Functionality

Create TM channel

- Click the + button at the top of the table to add a new TM channel.
- Input the required details.

Edit TC channel

- Click the *pencil icon* in the Functions column to modify TC channel details.
- Adjust the settings and save.

Attenuation

- Click *pencil outlined* in the row under attenuation. Modify value stored in the database for one particular pair of antenna and TC channel. Opens beacon-attenuation dialog.

Delete TC channel

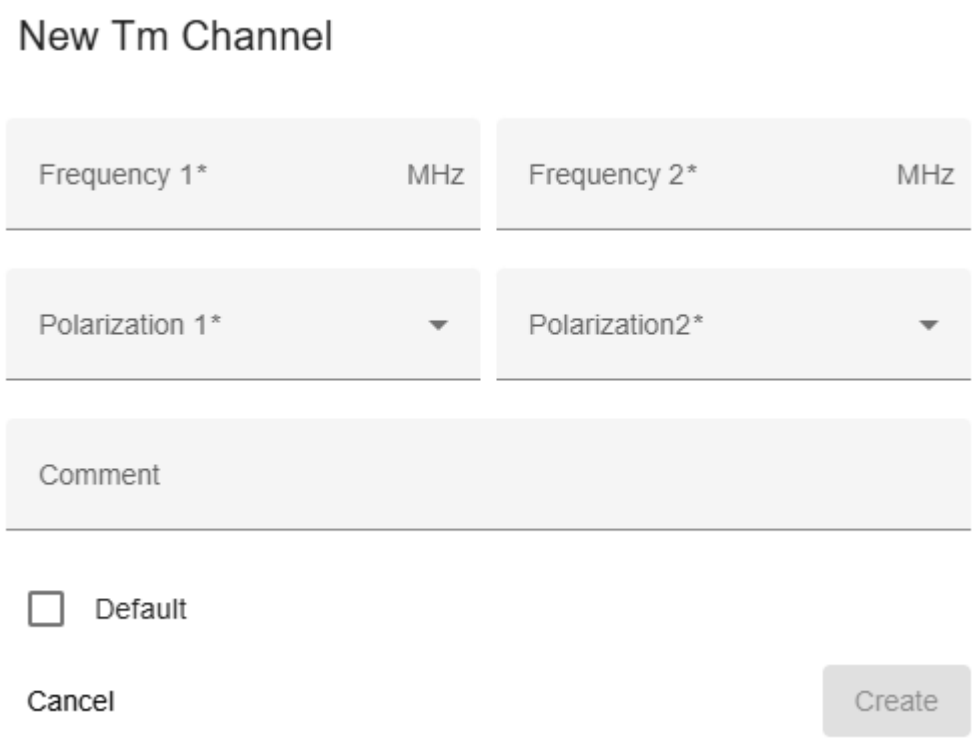
- Click the *trash icon* in the Functions column to delete an TC channel.
- Confirm the action in the pop-up dialog to delete the TC channel.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

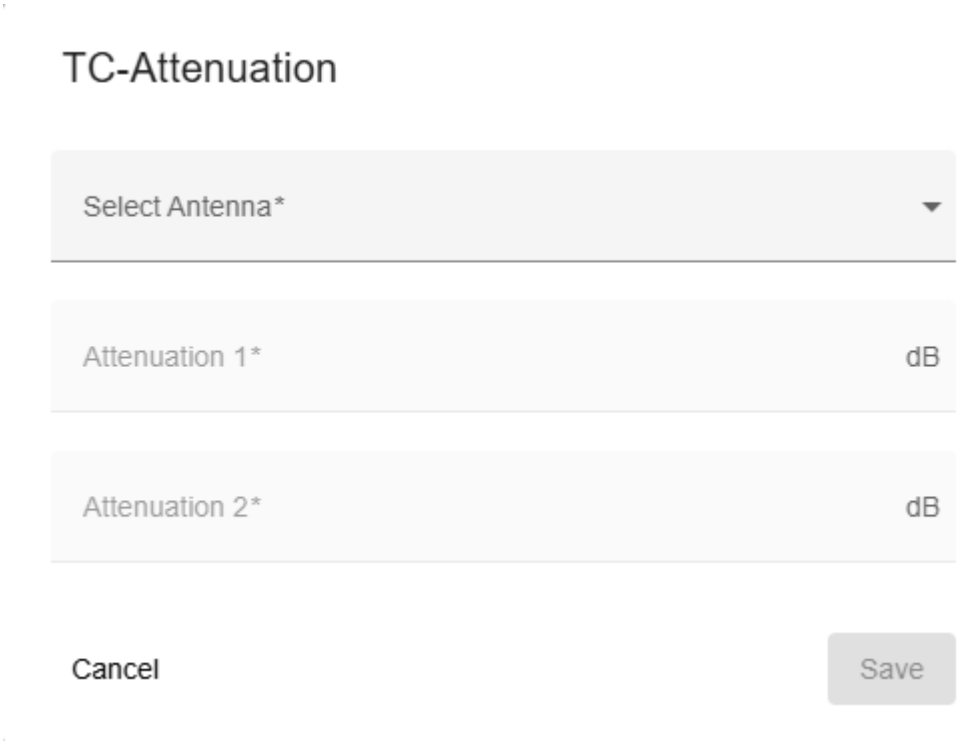
2.12.3.4.2 Dialogs

Add/edit TC channel dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit TC channel in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Frequency 1**: Primary receive frequency, floating point, MHz.
  - **Frequency 2**: Secondary receive frequency, floating point, MHz.
  - **Polarization 1**: This is an enumeration.
  - **Polarization 2**: This is an enumeration.
  - **Comment**: Free text describing the purpose of this dataset.
  - **Default**: Set TM channel as default.
- **Actions**:
  - **Create/Save**:
    - For new TM channel, this creates the TC channel with the provided details.
    - For existing TM channel, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

Modify TM channel attenuation



- **Trigger**: Click on the *pencil outlined* button in the row under *attenuation*
- **Fields**:
  - **Select antenna**: Drop-down to select antenna.
  - **Attenuation 1**: Transmit attenuation1, floating point, dB.
  - **Attenuation 2**: Transmit attenuation2, floating point, dB.
- **Actions**:
  - **Save**:
    - Saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

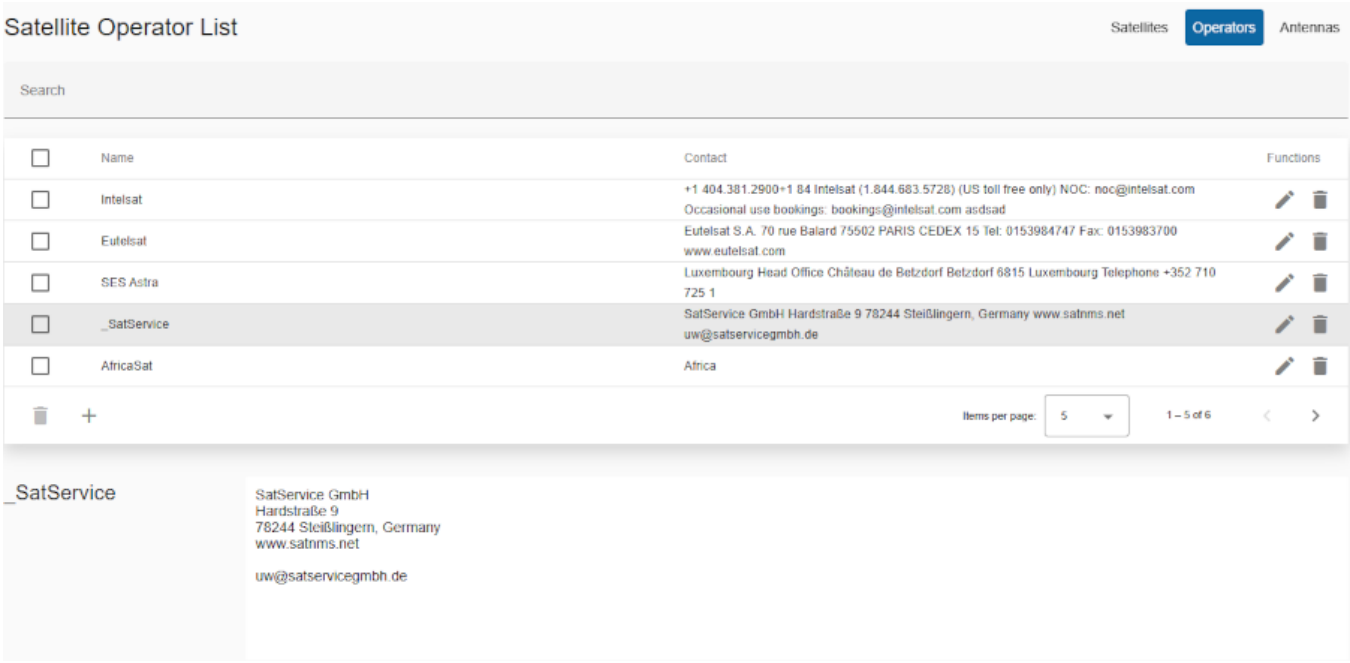
On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected TC channel.
- **Actions**:
  - **Yes**: Deletes the TC channel permanently.
  - **Cancel**: Cancels the deletion action.

2.13 Satellite operator list

All satellites operators known to system. You need to define a satellite operator before you can create an new satellite.

This list is also useful to store contact information of an satellite operator, e.g to perform a line-up before your start a transmission.



Title: Title of selected table (e.g. Operator list).

Menu options:

- **Satellites**: Displays the table of satellites.
- **Operators**: Displays the table of operators. The selected option has background color.
- **Antennas**: Displays the table of antennas.

Search bar

- Located at the top of the table.
- Allows users to filter events by keywords.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
  - **Location**: The checkbox column is located at the far left of the table.
  - **Purpose**: Allows users to select one or multiple rows for delete row(s).
  - **Functionality**:
    - **Single Selection**: Click on a single checkbox to select an individual row.
    - **Multiple Selection**: Click multiple checkboxes to select multiple rows at

once.
    - **Select all**: The checkbox in the header row allows you to select or deselect all rows on all pages.
    - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **Name**: The name of operator.
- **Contact**: Contact details of operator.
- **Functions**: Action icons for managing the operator:
    - **Edit**: Pencil icon to modify/edit the operator.
    - **Delete**: Trash can icon to delete the operator.

Rows: Clicking on operator row shows details of selected row (operator).

Bottom toolbar

- **Delete selected rows**: A button to delete selected rows from checkbox column.
- **Create**: + button is used to create new operator.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

Details of selected operator:

- Name of operator
- Contact details of operator

2.13.1 Functionality

Create operator

- Click the + button at the bottom of the table to add a new operator.
- Input the required details.

Edit operator

- Click the *pencil icon* in the Functions column to modify operator details.
- Adjust the settings and save.

Delete operator

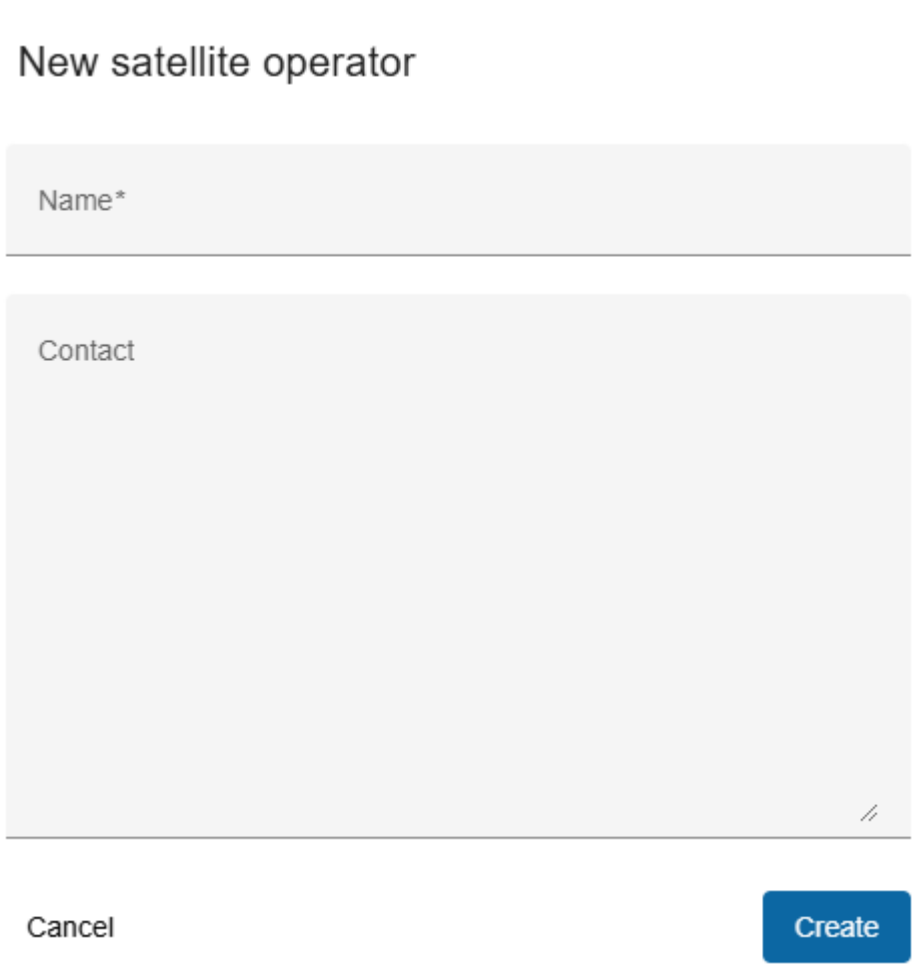
- Click the *trash icon* in the Functions column to delete an operator.
- Confirm the action in the pop-up dialog to delete the operator.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.13.2 Dialogs

Add/edit satellite dialog



- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new operator or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new operator, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Name**: The user defined name of this operator.
  - **Contact**: Contact details of operator.
- **Actions**:
  - **Create/Save**:
    - For new operator, this creates the operator with the provided details.
    - For existing operator, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

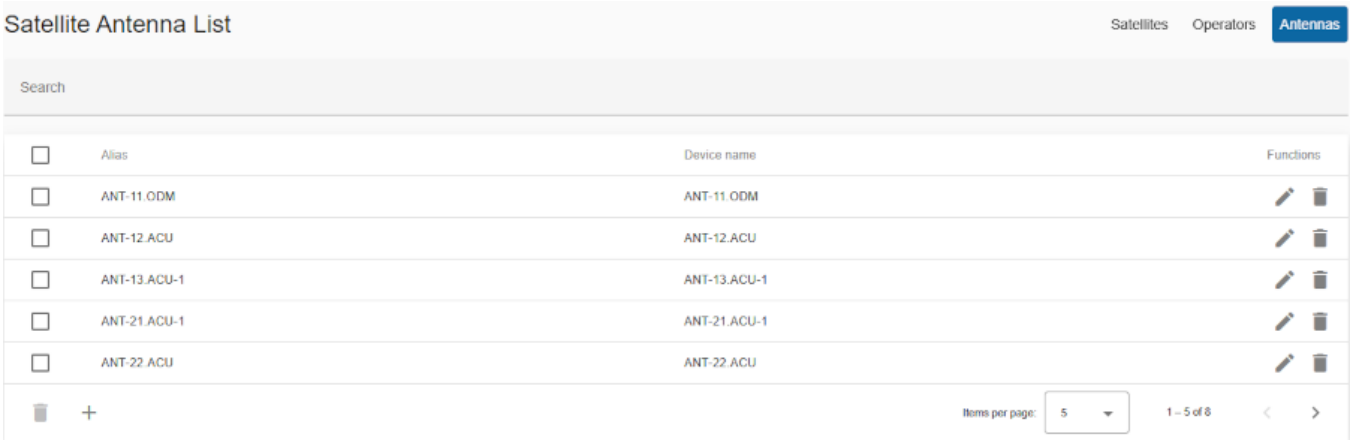
- **Trigger**: Opens when you click the trash icon for an operator from menu column or trash icon from bottom toolbar .
- **Content**: Asks the user to confirm whether they want to delete the selected operator(s).
- **Actions**:
  - **Yes**: Deletes the operator(s) permanently.
  - **Cancel**: Cancels the deletion action.

2.14 Antenna list

All saved antenna in database.

The position of a satellite (Azimuth, Elevation, Polarization) depends on the geographically location of the antenna and structural conditions of the antenna. Therefore a satellite position is different for every antenna.

With this list you define all antennas managed by this M&C system and which satellites are reachable with this antenna. To add a new satellite, your have to create the satellite before you can add here the position.



Title: Title of selected table (e.g. Antenna list).

Menu options:

- **Satellites**: Displays the table of satellites.
- **Operators**: Displays the table of operators.
- **Antennas**: Displays the table of antennas. The selected option has background color.

Search bar

- Located at the top of the table.
- Allows users to filter events by keywords.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.

- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
  - **Location**: The checkbox column is located at the far left of the table.
  - **Purpose**: Allows users to select one or multiple rows for delete row(s).
  - **Functionality**:
    - **Single Selection**: Click on a single checkbox to select an individual row.
    - **Multiple Selection**: Click multiple checkboxes to select multiple rows at once.
    - **Select all**: The checkbox in the header row allows you to select or deselect all rows on all pages.
    - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **Alias**: Alias of antenna.
- **Device name**: Device name from the MNC system.
- **Functions**: Action icons for managing the antenna:
  - **Edit**: Pencil icon to modify/edit the antenna.
  - **Delete**: Trash can icon to delete the antenna.

Rows: Clicking on antenna row shows other relatives tables like Satellite positions and Tracks of selected row.

Bottom toolbar

- **Delete selected rows**: A button to delete selected rows from checkbox column.
- **Create**: + button is used to create new antenna.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.14.1 Functionality

Create antenna

- Click the + button at the bottom of the table to add a new antenna.
- Input the required details.

Edit antenna

- Click the *pencil icon* in the Functions column to modify antenna details.
- Adjust the settings and save.

Delete antenna

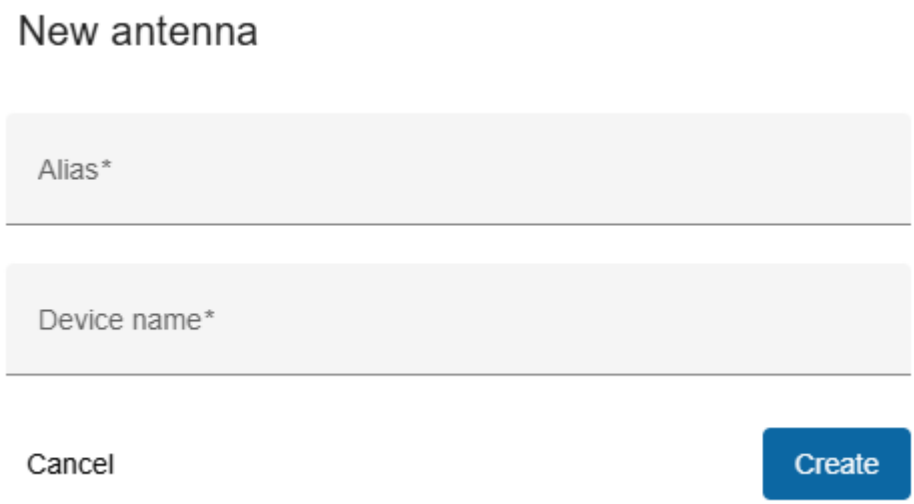
- Click the *trash icon* in the Functions column to delete an antenna.
- Confirm the action in the pop-up dialog to delete the antenna.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.14.2 Dialogs

Add/edit satellite dialog



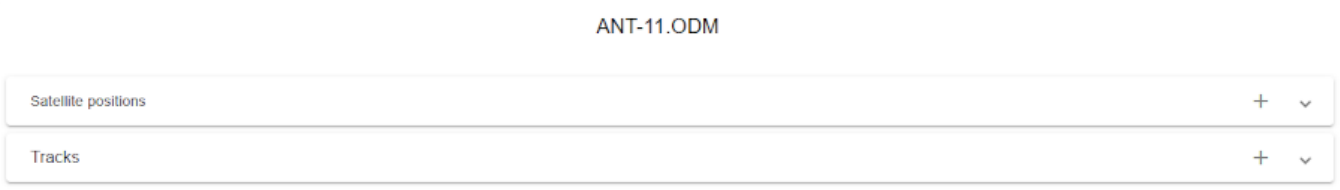
- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new antenna or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new antenna, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Alias**: Free text, can be used by user to add own name.
  - **Device name**: Free text, device name from the MNC system.
- **Actions**:
  - **Create/Save**:
    - For new antenna, this creates the antenna with the provided details.
    - For existing antenna, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an antenna from menu column or trash icon from bottom toolbar .
- **Content**: Asks the user to confirm whether they want to delete the selected antenna(s).
- **Actions**:
  - **Yes**: Deletes the antenna(s) permanently.

- **Cancel**: Cancels the deletion action.

2.14.3 Antenna related tables



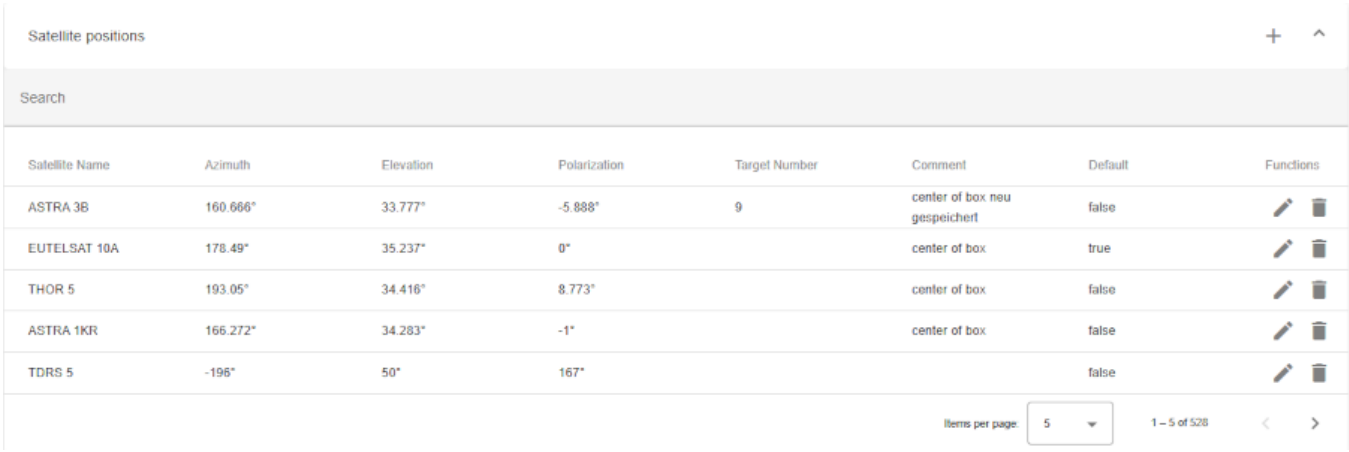
Title: Antenna name

Tables:

- Satellite position
- Tracks
- Import targets from Antenna Controller
- **Functionality**: Clicking on title or + or arrow to downside expands the table.

2.14.3.1 Satellite position table

The list of all satellite positions for the given antenna.



Top

- Left-side title of table.
- Plus (+) button to create new satellite position.
- Icon button to expand or collapse table

Search bar

- Located at the top of the table.
- Allows users to filter events by keywords.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

Accessing the table from satellites page:

- **Device name**: Device name from the MNC system.
- **Alias**: Alias of antenna.

Accessing the table from antennas page:

- **Satellite name**: The name of the satellite to which this data set relates.

Common columns:

- **Azimuth**: Antenna azimuth pointing angle in degrees (floating point).
- **Elevation**: Antenna elevation pointing angle in degrees (floating point).
- **Polarization**: Antenna polarization pointing angle in degrees (floating point).
- **Target number**: Saved target number at antenna controllers.
- **Comment**: Text describing the purpose of this database entry.
- **Default**: Default position for selected antenna. Excluded from satellite page
- **Functions**: Action icons for managing the satellite position:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.14.3.1.1 Functionality

Create satellite position

- Click the + button at the top of the table to add a new satellite position.
- Input the required details.

Edit satellite position

- Click the *pencil icon* in the Functions column to modify satellite position details.
- Adjust the settings and save.

Delete satellite position

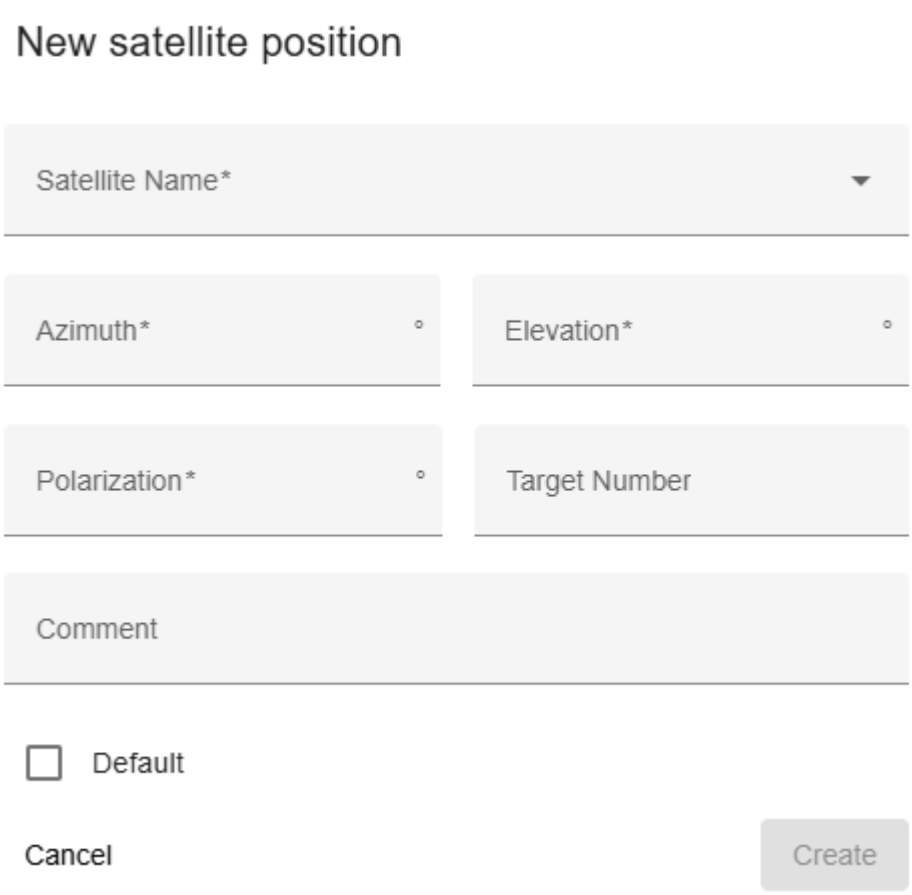
- Click the *trash icon* in the Functions column to delete an satellite position.
- Confirm the action in the pop-up dialog to delete the satellite position.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.14.3.1.2 Dialogs

Add/edit satellite position dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit satellite position in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Satellite name**: Select an satellite name from drop-down to which the data set should refer.
  - **Filename**: The name of the data file containing the position data of the satellite.
  - **Comment**: Free text describing the purpose of this dataset.
  - **Default**: Set position as default.

    - **Functions**: Action icons for managing the track:
        - **Edit**: Pencil icon to modify/edit.
        - **Delete**: Trash can icon to delete.

- **Actions**:

    - **Create/Save**:
        - For new position, this creates the satellite position with the provided details.
        - For existing position, this saves modifications.
        - If an error occurs in the input fields, the button is deactivated.
    - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an satellite position from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected satellite position.
- **Actions**:
    - **Yes**: Deletes the satellite position permanently.
    - **Cancel**: Cancels the deletion action.

2.14.3.2 Tracks table

Especially for fast moving satellites or satellites in a transition phase it may be necessary to move the antenna with a pre-calculate list of satellite postions. These list is called a track. Files with tracks are usually created by satellite operators. These files have one line per step with: timestand, azimuth, elevation and polarization.

Top

- Left-side title of table.
- Plus (+) button to create new track.
- Icon button to expand or collapse table

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon

near the column name.

Table columns

- **Satellite name**: The name of the satellite to which this data set relates.
- **Filename**: Saved name of the data file containing the position data of the satellite.
- **Comment**: Text describing the purpose of this database entry.
- **Default**: Default track for selected antenna.
- **Functions**: Action icons for managing the track:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.14.3.2.1 Functionality

Create track

- Click the + button at the top of the table to add a new track.
- Input the required details.

Edit track

- Click the *pencil icon* in the Functions column to modify track details.
- Adjust the settings and save.

Delete track

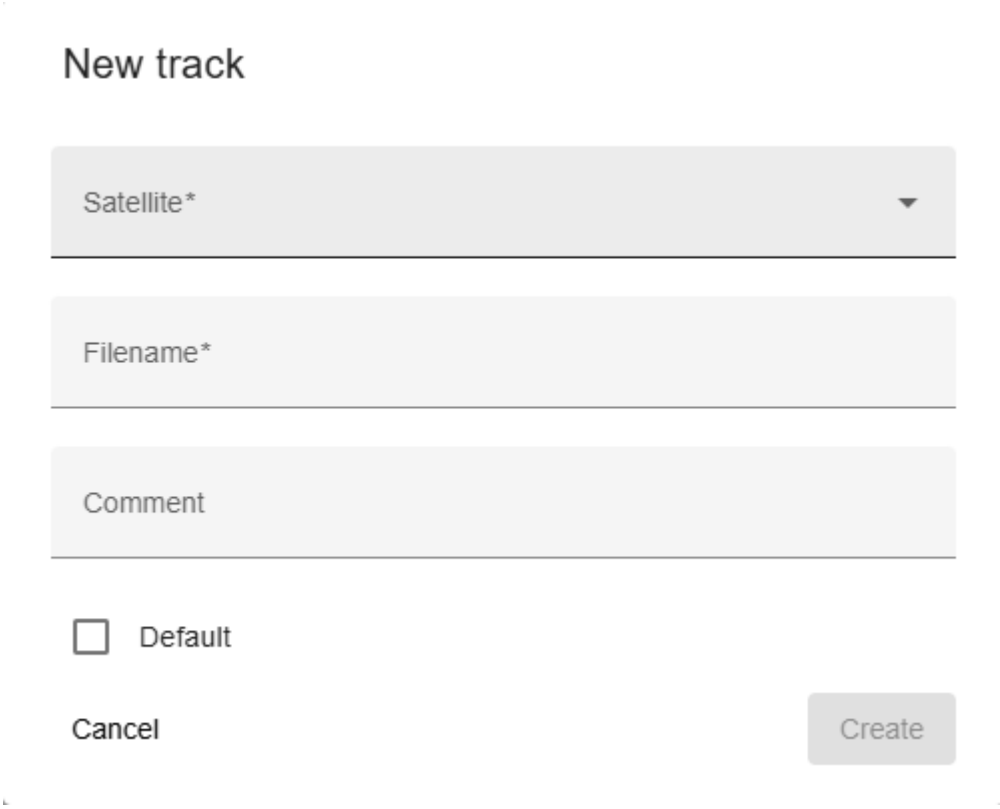
- Click the *trash icon* in the Functions column to delete an track.
- Confirm the action in the pop-up dialog to delete the track.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.14.3.2.2 Dialogs

Add/edit track dialog



- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit track in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Satellite name**: Select an satellite name from drop-down to which the data set should refer.
  - **Filename**: The name of the data file containing the position data of the satellite.
  - **Target number**: A target number to be set at some antenna controllers.
  - **Comment**: Free text describing the purpose of this dataset.
  - **Default**: Set track as default.
- **Actions**:
  - **Create/Save**:
    - For new track, this creates the track with the provided details.
    - For existing track, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an track from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected track.
- **Actions**:
  - **Yes**: Deletes the track permanently.

- **Cancel**: Cancels the deletion action.

2.14.3.3 Imports table

This tables contains the target list read from an ACU device.

Top

- Left-side title of table.
- Icon button to expand or collapse table

Search bar

- Located at the top of the table.
- Allows users to filter events by keywords.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Satellite name**: The name of the satellite to which this data set relates.
- **Azimuth**: Antenna azimuth pointing angle in degrees (floating point).
- **Elevation**: Antenna elevation pointing angle in degrees (floating point).
- **Polarization**: Antenna polarization pointing angle in degrees (floating point).
- **Target number**: Saved target number at antenna controllers.
- **Functions**: Action icons for managing the ACU-Target:
  - **Import**: Open a dialog to perform action.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and

navigate between pages.

2.14.3.3.1 Functionality

Import satellite position

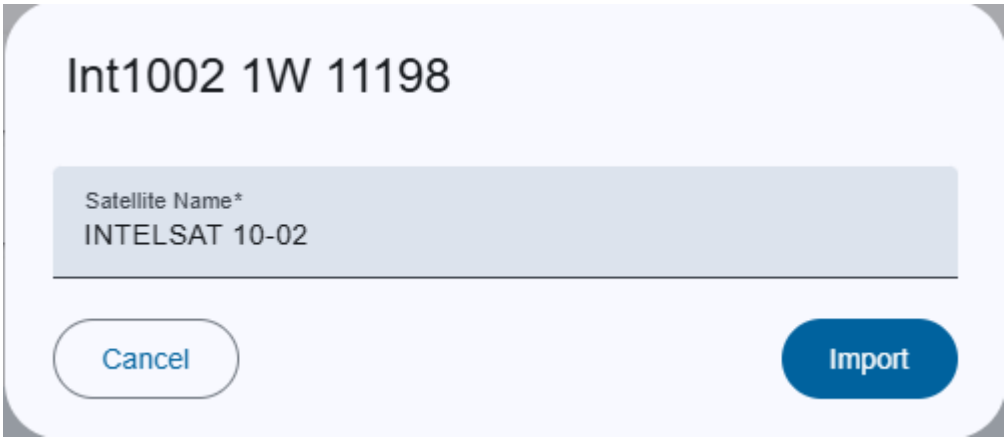
- Click the icon opens a dialog to import values.

Pagination

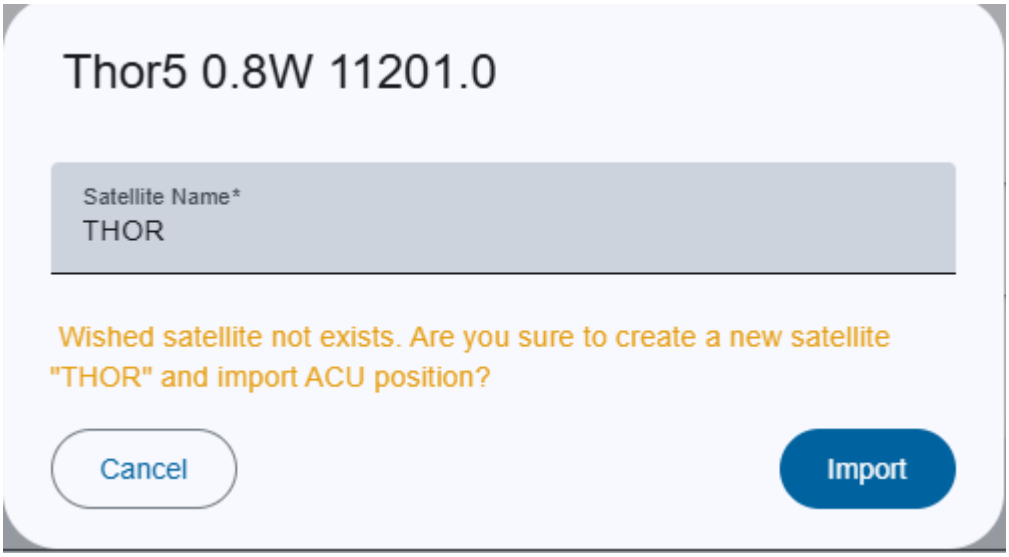
- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.14.3.3.2 Dialogs

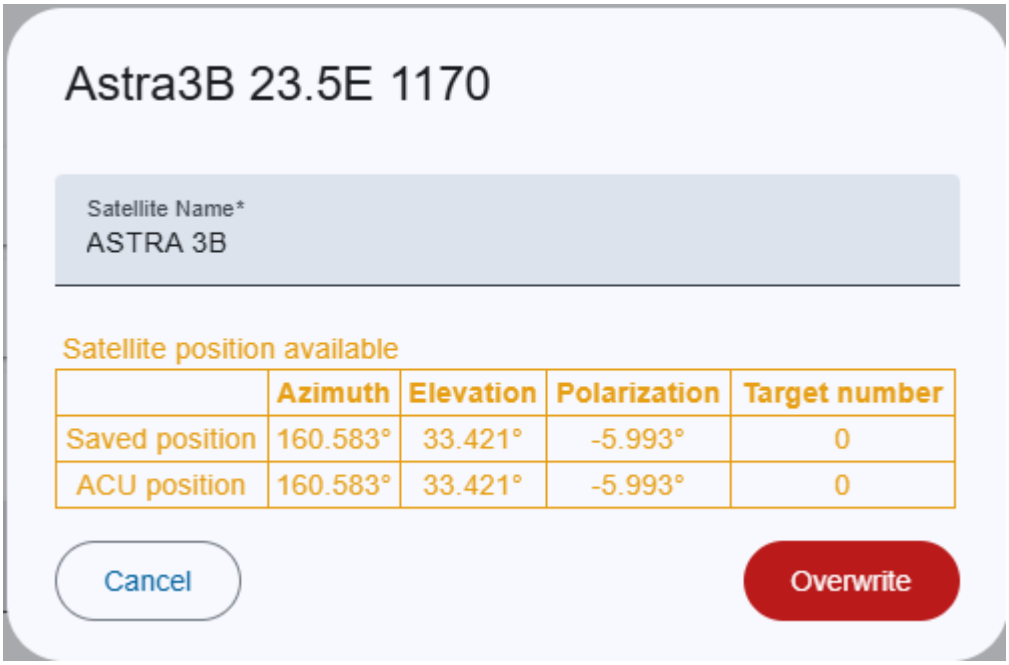
Import satellite position



- **Trigger**:
  - when you click on the icon to import values
  - The dialog title clearly indicates the name of "Satellite name" saved on ACU Target list
- **Fields**:
  - **Satellite name**: Select an satellite name from drop-down to which the data set should refer.
  - **Warnings**:
    - If the specified satellite is not saved



- If the satellite position is exists, the values of both are displayed to show the difference.



- **Actions**:
  - **Import/Overwrite**:
    - For new satellite or satellite position, this creates the satellite or satellite position with ACU-Target values.
    - For existing position, this overwrites with ACU-Target values.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

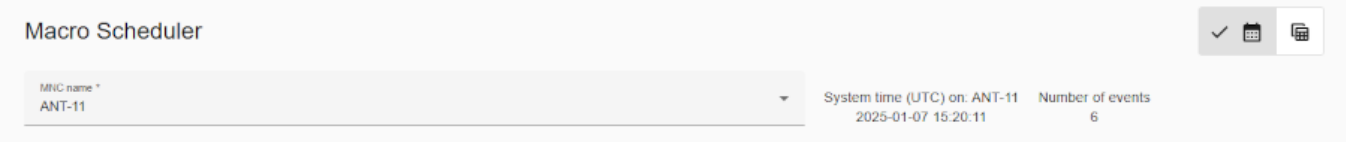
2.15 The Macro Scheduler

The Macro scheduler included in the sat-nms M&C software lets you execute macros at planned times. You access the macro scheduler from the main window by clicking to the toolbar icon which shows a alarm icon.

The Macro scheduler supports events which are executed once at a given time as well as repeated events on a daily, weekly, monthly or yearly base. Repeated events are executed in the planned was up to a given end date. Monthly repeated events are executed every month at the programmed day of month. If you program a monthly event for 29.-31. of a month, it will be executed at the last day of a month which is shorter.

The Macro scheduler page shows a list (in calender- or table-view) of planned events and provides controls to add, edit or delete events. Like with the macro/event management window, you may either select the recorded/stored macro you want to be played at the event's time or enter the commands manually.

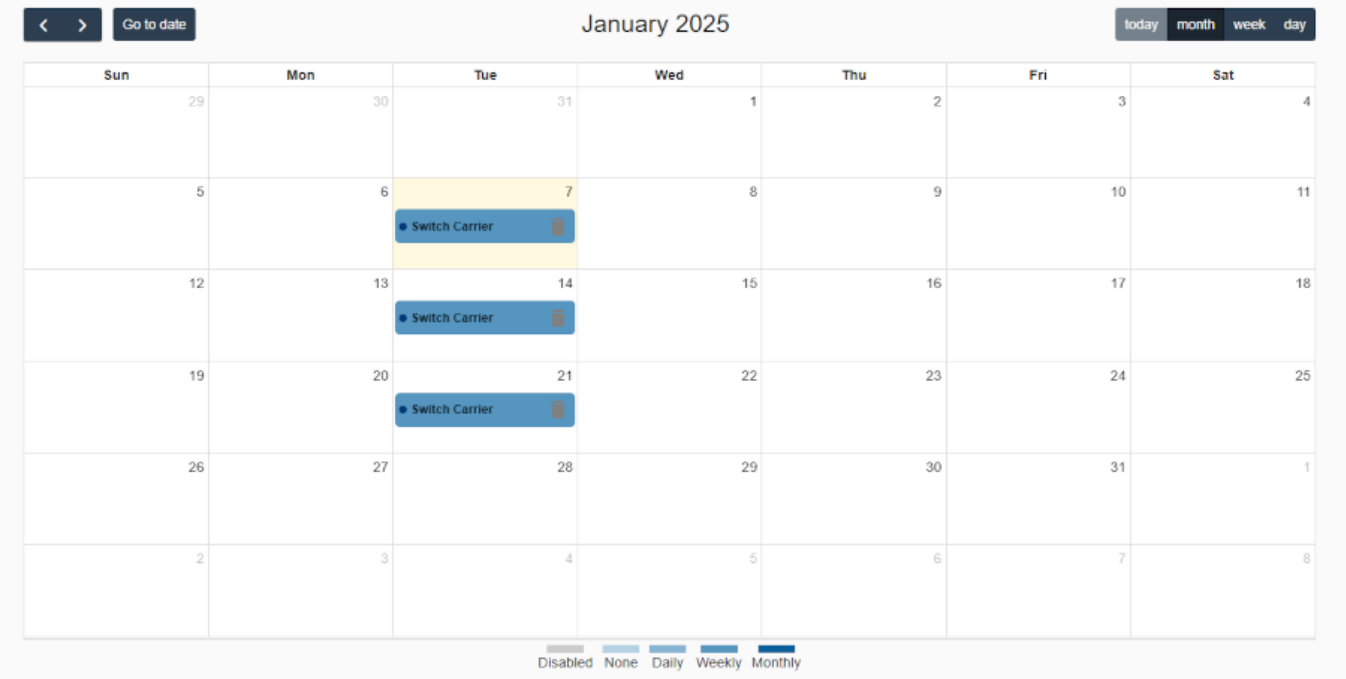
2.15.1 Scheduler



1. **Title**: Title of scheduler page.
2. **View options**: At the top right of the page are view options.
   - Calendar view (default)
   - Table view
3. **Select an MNC**: Below the heading and the view option, you must select an MNC from the drop-down list.
4. **Current system time**: After selecting the MNC near the drop-down list, the current system time in UTC is displayed.
5. **Total events**: Total number of events of the selected MNC is displayed.

2.15.2 Calender view

The Calendar View provides an interactive interface to visualize, create, manage, and track events based on different time frames (day, week, month). Users can navigate between dates, create new events, edit existing ones, reschedule an event by drag and drop, and distinguish events using a color-coded legend.



Navigation bar (top)

- **Left and right Arrows**: Use these arrows to navigate to the previous or next time-frame (month, week, or day, depending on the selected view).
- **Go to date button**: Click to open a dialog where you can select a specific year and month to navigate directly.

Current date

- In the middle, there is a heading displaying the current month and year. If you select a view other than the month, it will show the date range for the week or, for the day view, the specific month, day, and year.

View selection buttons

- Located in the top-right corner:
    - **Today**: If the current date is already displayed, the button is disabled. Otherwise, clicking the button takes you directly current date.
    - **Month**: Displays the calendar in a monthly view (default).
    - **Week**: Switches the calendar to a weekly view.
    - **Day**: Displays the calendar in a daily view.

Calender

- In the middle is the calendar view, displaying the selected view. Clicking on an event allows you to edit it, and clicking on the delete icon lets you delete the event. If you click on an empty date, you can create a new event. You can also reschedule an event by dragging and dropping it to a different date. A pop-up will appear to confirm the changes.
- Once an event is scheduled, the calendar will display it only on the set date. If the event

is set to repeat weekly, it will appear every week until it expires, and similarly for yearly or monthly or daily recurring events.

Color-coded legend (bottom)

- Clarification of the calendar colors, explaining what each color represents.

2.15.2.1 Features

Create event

- Click on an empty date to open the event creation dialog.
- Specify event details, including title, start date, end date, and recurrence type.

Edit event

- Click on an event to modify its details.
- Adjust the event date, recurrence, or other settings.

Delete event

- Use the trash icon within an event to delete it.
- Confirm the action in the pop-up dialog to delete the event.

Drag and drop rescheduling

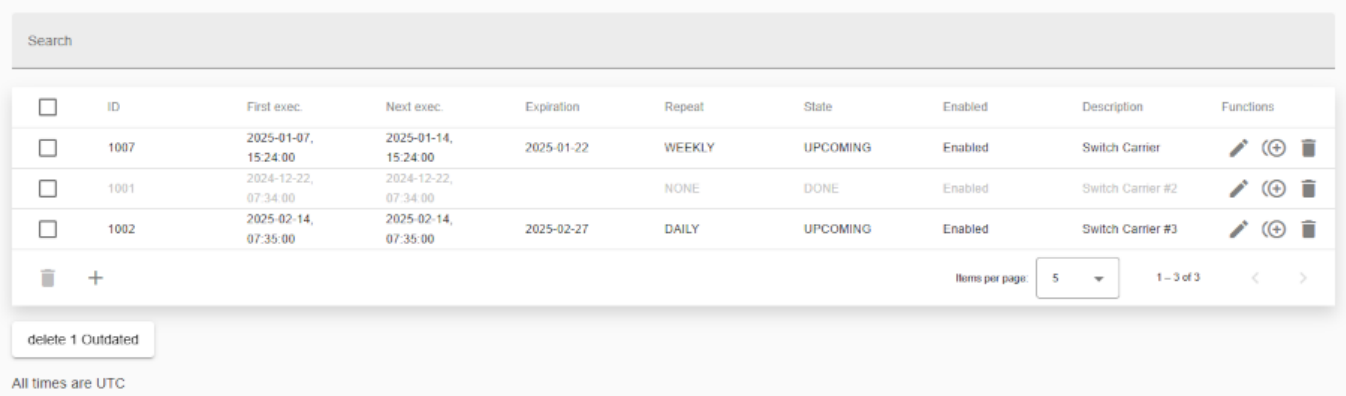
- Drag an event from one date to another to reschedule it.
- A confirmation dialog will appear to validate the changes.

Time frame navigation

- Navigate seamlessly using the arrows or the *Go to Date* button to jump to specific months, weeks, or days.

2.15.3 Table view

The Scheduler Table View provides a detailed interface to manage and monitor scheduled events. It allows users to view, edit, delete, and create events efficiently. All times displayed are in UTC.



Search bar

- Located at the top of the table.
- Allows users to filter events by keywords, such as ID, description, repeat type, etc.

Header row

- You can sort rows in ascending or descending order by clicking on the column header (e.g., ID, First Exec., Next Exec., etc.).
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
  - **Location**: The checkbox column is located at the far left of the table.
  - **Purpose**: Allows users to select one or multiple events for delete event.
  - **Functionality**:
    - **Single Selection**: Click on a single checkbox to select an individual event.
    - **Multiple Selection**: Click multiple checkboxes to select multiple events at once.
    - **Select All**: The checkbox in the header row allows you to select or deselect all events on all pages.
    - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **ID**: Unique identifier for each scheduled event.
- **First exec.**: The date and time when the event is scheduled to execute for the first time.
- **Next exec**.: The next scheduled execution date and time for the event.
- **Expiration**: The date when the event is set to expire and stop repeating.
- **Repeat**: Indicates the recurrence type of the event:
  - **NONE**: The event does not repeat.
  - **DAILY**: The event repeats every day until expiration.
  - **WEEKLY**: The event repeats weekly until expiration.
  - **YEARLY**: The event repeats yearly until expiration.
- **State**: Displays the current state of the event:
  - **UPCOMING**: The event is scheduled for a future execution.
  - **DONE**: The event has been executed.
- **Enabled**: Shows whether the event is active or disabled.
- **Description**: A brief description of the event.
- **Functions**: Action icons for managing the event:
  - **Edit**: Pencil icon to modify the event.
  - **Delete**: Trash can icon to remove the event.
  - **Duplicate**: Circular arrow icon to duplicate the same event.

Bottom toolbar

- **Delete selected events**: A button to delete selected events/rows from checkbox

column.
- **Create event**: + button is used to create new event.
- **Delete outdated**: A button to delete outdated events from the list.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.
- **UTC notice**: A reminder that all times displayed are in Coordinated Universal Time (UTC).

2.15.3.1 Features

Create event

- Click the + button at the bottom of the table to add a new event.
- Input the required details, such as execution time, expiration, recurrence type, etc..

Edit event

- Click the *pencil icon* in the Functions column to modify event details.
- Adjust the settings and save.

Duplicate event

- Click the *circular arrow* icon to duplicate an event.
- A dialog will appear with same settings (you can change some settings too).
- Click on save to preform action

Delete event

- Click the *trash icon* in the Functions column to delete an event.
- Confirm the action in the pop-up dialog to delete the event.

Delete outdated events

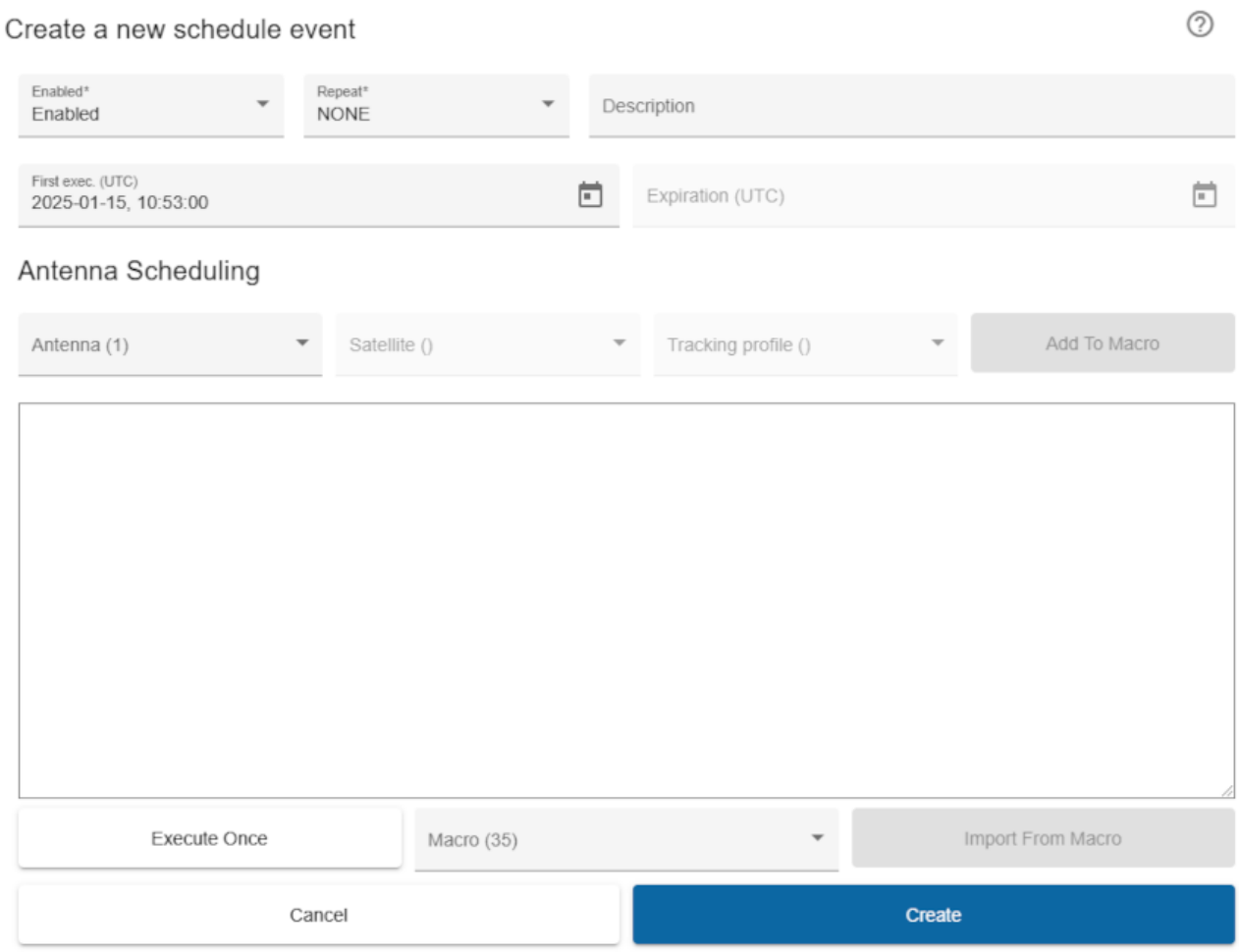
- Use the *Delete Outdated* button to remove all events that are no longer relevant or expired.
- Number shows that how much outdated events are there.
- A confirmation dialog will appear to with outdated event(s) details to perform action.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.15.4 Unified dialogs

Schedule-event dialog



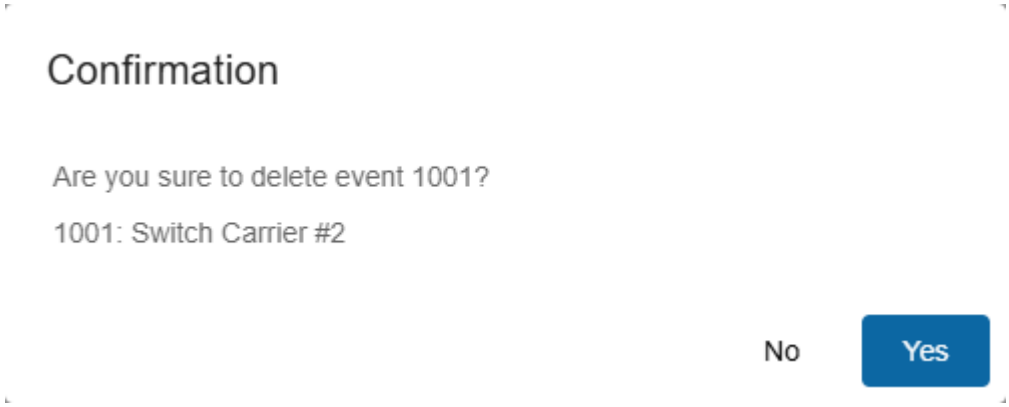
- **Trigger**:
  - when you click on an empty date in the Calendar View or the "+" button in the Table View (to create a new event) or
  - an existing event (to edit) in either the Calendar View on click on an event or in the Table View on *pencil icon* (to edit) or *circular arrow* (to duplicate) in the Functions column
  - The dialog title clearly indicates whether you are creating a new event, editing or duplicating an existing one, ensuring clarity in the action being performed.
  - There is a help button on the right-hand side that leads directly to the scheduler documentation.
- **Fields**:
  - **Enabled**: If enabled, the event is enabled. An enabled event will execute its macro at due time, a disabled event will skip this
  - **Repeat**: The repeat mode for the event. This may be one of:
    - **NONE**: The event is executed once at the date and time specified in "first execution".

        - **DAILY**: The event is executed first at the date and time specified in "first execution", then at all following days at the same time but not beyond the day given in "last execution" below.
        - **WEEKLY**: The event is executed first at the date and time specified in "first execution", then every seven days at the same time but not beyond the day given in "last execution" below.
        - **MONTHLY**: The event is executed first at the date and time specified in "first execution", then every month at the same day of month and time but not beyond the day given in "last execution" below. If you program a monthly event for 29.-31. of a month, it will be executed at the very last day of any month which is shorter.
    - **First exec. (UTC)**: The date and time of the first execution of the event (for repeated events) or simply the time of execution for an event executed only once. Click into the calender icon in field to open a pop up calendar widget which assists you entering the date. *NOTE: You have to enter time manually.*
    - **Description**: A brief description of the event.
    - **Expiration (UTC)**: The day of the last execution of the event.
    - **Antenna scheduling**: The Macro Scheduler provies a function to create easily a macro to start tracking at avert in antenna. The antenna scheduler function is only available for antennas which are managed by Antenna-Management devices. If there are no such devices configured in the M&C, the fields assigned to the antenna scheduler function appear disabled.
        - **Antenna**: Select an antenna from drop-down.
        - **Satellite**: Select an satellite from drop-down.
        - **Tracking profile**: Select an tracking profile from drop-down.
        - **Add to macro**: Replaces the macro text with a fragment which sets up the antenna for the selected satellite and tracking mode and starts tracking on it. Additional commands / parameter settings may be added to the macro (below this button in text field) before the schedule event gets saved.
        - **Macro text**: Brief description of an macro to be saved. It is also editable.
    - **Macro**: Select an recorder macro from drop-down list.

- **Actions**:

    - **Execute Once**: To execute the macro once. Useful for testing the macro.
    - **Import From Macro**: It imports the selected macro and displays the details in the free area above it.
    - **Create/Save**:
        - For new events, this creates the event with the provided details.
        - For existing events, this saves any modifications.
    - **Cancel**: Closes the dialog without saving.

On delete event confirmation dialog

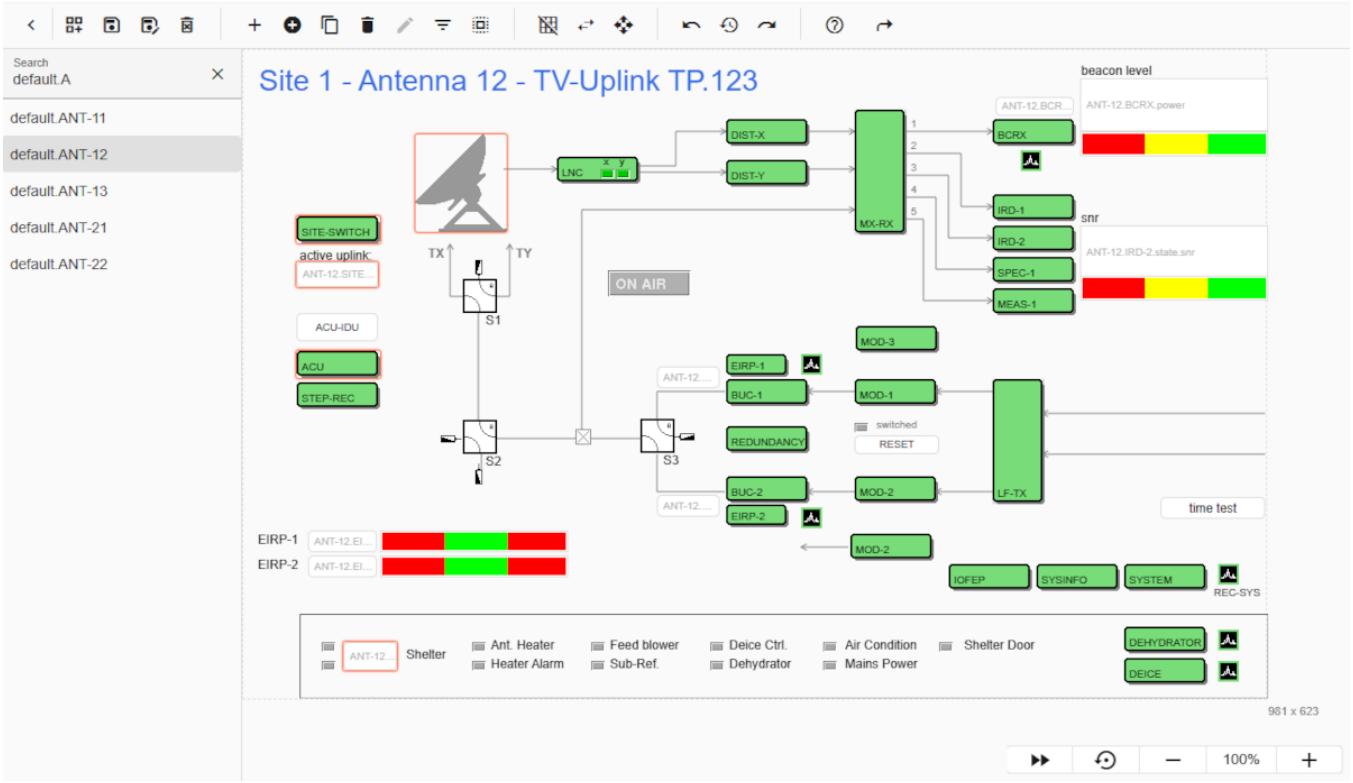


- **Trigger**: Opens when you click the trash icon for an event in either view or click delete outdated in table view.
- **Content**: Asks the user to confirm whether they want to delete the selected/outdated event(s).
- **Actions**:
  - **Yes**: Deletes the event(s) permanently.
  - **Cancel**: Cancels the deletion action.

2.16 Screen editor

The Screen editor is used to layout or configure the M&C windows called user screens used in the M&C user interface.

The Screen Editor works like a simple drawing program. Each M&C user screen contains a number of objects (elements) which may be placed, sized and edited using the screen Editor.

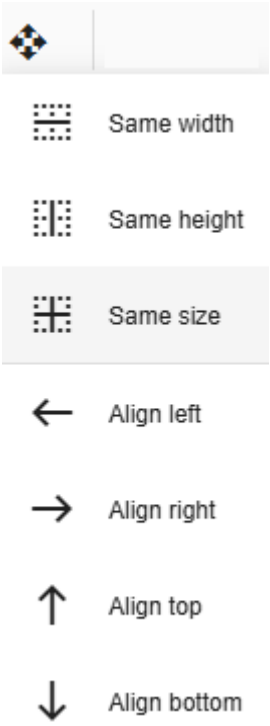


Toolbar

The toolbar options and their functions are described below. If you move the mouse pointer over the icon, the tooltip is displayed with a brief description.

- **Toggle screen list**: Clicking on arrow to left (<, >) toggle the screen list.
- **Create new screen**: Opens a create new screen dialog.
- **Save screen**: Save the modifications.
- **Save screen as**: Save the selected screen under other name.
- **Delete screen**: Delete the selected screen.
- **Add element**: Opens a panel on the right side to add an element.
- **Insert screen**: This option is used to insert an other user screen elements to selected screen.
- **Duplicate element(s)**: Duplicates the selected element(s) with there properties. The new element(s) appears on top of existing one.
- **Delete element**: Deletes the selected element(s). You can also use backspace or delete key to perform this action.
- **Edit element**: Opens a element properties panel on right side of selected element. You can also use double click to edit element.

- **Select element type**: This option is used to select all elements according to their type. If you click on this option, an input field would be displayed. When you start typing, the element list is filtered and you can select an option. All elements with the type you have selected from the option will be selected.
- **Select all elements**: Select all elements. You can also use `ctrl + a` .
- **Grid positioning**: Elements are placed along a invisible 3 x 3 pixel grid if they are moved while this options in on.
- **Push element to background**: Moves the selected element to the back so it does not cover any other element.
- **Alignment**: Aligning a group of elements. The first selected element is the parent element, i.e. the properties are transferred from the parent element to the child elements.



  - **Same width**: It is used to have the same width as the parent element.
  - **Same height**: It is used to have the same height as the parent element.
  - **Same size**: It is used to have the same width and height as the parent element.
  - **Align left**: Align the elements to the left of parent element.
  - **Align right**: Align the elements to the right of parent element.
  - **Align top**: Align the elements to the top of parent element.
  - **Align bottom**: Align the elements to the bottom of parent element.
- **Undo**: To go to previous change. Can also be used with `ctrl + z` .
- **History**: All changes made by a user are saved in the history. If you click on the clock

symbol, a list of all changes is displayed with the number and date-time (1: yyy-mm-dd HH:mm:ss). If you click on an option, you will be taken to this specific change.

- **Redo**: To go to forward change. Can also be used with `ctrl + y` .
- **Help**: Bring you to help page of screen-editor.
- **Shortcuts**: List of all key combinations.

Screen List

List of all saved screens on the server. Select the name of the screen you wish to edit.

Editing area

After selecting the screen, the screen elements are loaded into the editing area on the right-hand side like in image above. If you look closely, some elements have a *light red* border, which means they are selected. You can simply start dragging, resizing, etc. to perform functions. The functionality is described next in functionality section.

- There is a small toolbar on the bottom right-hand side of the editor. The functions are there:
  - **Jump to user-screen**: You can go direct jump to the user-screen.
  - **Reset position**: If you click on this button, the position of the screen editor is reset.
  - **Zoom out**
  - **Current zoom**: Displays the current zoom of the screen monitor. Clicking on the number restores the standard zoom (100%).
  - **Zoom in**

2.16.1 Functionality

Double click: Double click on an element to edit the element properties.

Resize element: Once you have selected the element, you can change the size of the element in any direction from the corner of the element. If you move the mouse over it, the mouse pointer changes from the standard pointer to the corresponding direction pointer.

Select multiple elements: Just hold down hte `ctrl` key and click on that element you want to select.

Reposition element(s): Once you have selected an element(s), you can drag it to the desired location and drop it there.

Edit element properties: You can edit the element properties in the *Element properties* panel. This is displayed on right hand side of editor when you double-click on the element or click *pencil icon* to edit the selected element in the toolbar.**Duplicate element(s)**: Duplicates the selected element(s) with there properties. The new element(s) appears on top of existing one. **Delete element**: Deletes the selected element(s). You can also use backspace or delete key to perform this action. **Zooming**:

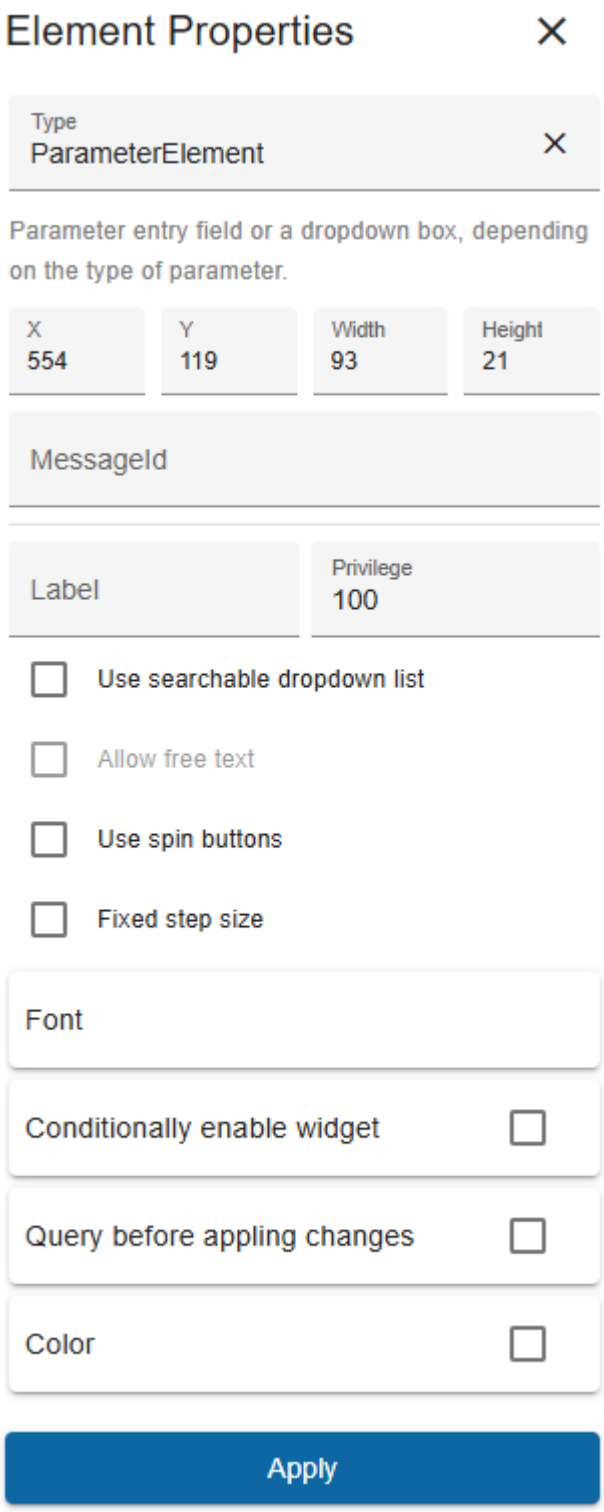
- **Key combinations**
  - **Zoom in**: By hold down `ctrl` key and pressing `+` key.
  - **Zoom out**: By hold down `ctrl` key and pressing `-` key.
  - **Default zoom (100%)**: By hold down `ctrl` key and pressing `0` key.
- **Mouse and key combination**: Simply hold down `ctrl` key and scroll with `mouse wheel`

Move screen: By hold down `space` key and hold down `mouse click` and then move the mouse, the screen will move in the direction in which the mouse is moving.

Horizontal scroll: By hold down `shift` key and scroll with `mouse wheel`

2.16.2 Element properties

The window for the properties of a screen element is used to define the properties of a display element within the screen editor. The screen editor uses the same window with the element properties for all elements. If you click on another element in the editor window, the available properties of this element are loaded into the properties window, which may contain different fields/information depending on the type of screen element. The element properties for the general *parameter element* are shown below as an example:



Each element has the properties type, x, y, width and height.

1. **X**: X location of the element in the user screen.
2. **Y**: Y location of the element in the user screen.
3. **Width**: Width of the element's bounding box.

4. **Height**: Height of the element's bounding box.
5. **Type**: The type of the element. When you add a new element, ParameterElement is the default value. When editing an element, the type of element will be selected element type.

NOTE: All sections like *Font or Color* are expandable sections. Just click on that section to expand or collapse.

2.16.2.1 Display element types and their properties

The table below shows the types of display elements the screen editor supports. A detailed description of the individual element types is given on the following pages.

Type	Description
Arrow element	Draws a horizontal/vertical line or arrow.
AzEl element	A chart showing the tracking history of an antenna controller as a point cloud in an elevation over azimuth coordinate system.
Button element	Defines a button which launches another screen, e.g. for detail views.
Chart element	A chart element, displaying a numeric parameter as an y/t diagram that automatically advances with 1 pixel / second.
Device icon element	Places an icon into the screen which represents a device. This icon will display the device's operating/fault state by it's color / shape.
Display element	A parameter display field in which the parameter cannot be edited.
Frame element	Draws a sunken 3D frame, may be used to group parameters.
Gauge element	A gauge element, displaying a numeric parameter as a horizontal bar.
Icon element	Places an icon (GIF or JPG image) into the screen. Optionally the icon can be programmed to change with a parameter value.
Latching button element	A button which displays/controls a 2-state parameter using it's pressed state.
Parameter button element	A button which sends a certain parameter value when pressed.

Type	Description
Parameter element	A parameter entry field.
Radio button element	A parameter entry field specially for **Choice** parameters.
Rect element	Draws a rectangle.
Spectrum element	The spectrum display element integrates the spectrum display of a spectrum analyzer device in a user screen.
Switch element	Like the device icon. but additionally displays the actual position of a switch (Meant to be used for block diagrams showing the true signal path).
Target list element	The ODM Target List element shows the list of targets of a SatService-ACU-ODM antenna controller device. It permits to recall, save or delete target definitions of this type of antenna controller. It is specialized to this antenna controller, does not support other types.
Text element	Displays a line of text.
Thumbnail icon element	A screen element to show thumbnail images from the video processed by an encoder / decoder / gateway device which supports this feature.
XY chart element	This element shows the relation of two numeric variables in an X/Y diagram, featuring a trace which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

2.16.2.1.1 Button element

The button (frame) display element creates a button which launches another user interface screen if pressed. The screen to be launched may be another user defined screen or a predefined one.

The attributes listed below may be configured to customize a display element of this type:

- **Label**: The label text is drawn on the button.
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.

- **Use icon instead of label**: Check this checkbox in order to make the button show an icon instead of a label text.
- **Icon**: Provides a list of all available icons / images on the sat-nms server.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Button action**: with this field you select the function of the button. The available functions are listed below. The meaning of the parameters right beside the function selector changes with the function selected.
  - **BROWSER-VIEW**: Launches the platform's standard web browser and shows the given URL. URL must be entered.
  - **CHILD-SCREEN**: Launches a new screen in a separate window. Screen name must be entered.
  - **FREC-VIEW**: Launches a graphical presentation screen for data recorded with a File-Recorder logical device. MNC.device name must be provided (e.g. ANT-11.REC-SYS).
  - **LOAD-PRESET**: Launches a dialog to select and apply a preset to a given device. MNC.device name must be provided (e.g. ANT-11.REC-SYS). You can also set driver and search pattern. Search pattern field with this value. Only presets containing the pattern string (character case doesn't matter) will appear in the list.
  - **REPLACE-SCREEN**: Replaces the actual screen by another one, re-uses the same window.
  - **SPECTR-VIEW**: Launches the device window of a spectrum analyzer and optionally sets an arbitrary number of parameters. MNC.device name must be provided (e.g. ANT-11.REC-SYS). *An optional macro list*: This may contain zero or more parameter ID - value pairs. These settings may be used to preset the spectrum analyzer certain settings but also to control switches or other devices in the signal path. Example: S03=B, SA1.center=12345.666 If the operator clicks to this button, the switch S03 is commanded to position B, the spectrum analyzer SA1 is set for a center frequency of 12345.666 MHz and the spectrum analyzer window gets opened.
  - **TRACK-VIEW**: Launches a tracking report window for the Antenna-Tracking logical device. The tracking report window shows azimuth or elevation together with the beacon level over the recent 48 hours. MNC name and ACU driver must be provided. Typically this is the "state.mode4" Parameter of an Antenna-Tracking or a SatService-ACU-ODM device.
  - **TREE-NAVIGATE**: This button used to navigate to tree devices as user screen. Tree path must be MNC.Subsystem (e.g. ANT-11.GATES)

- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.

    - **Enable ID**: This field defines the ID for the enabling parameter.
    - **Enable value**: The value belonging to the parameter above. The actual value of the parameter

- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.

    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.2 Parameter element

The *Parameter* display element is the common component to display and edit most types of M&C parameters. Depending on the data type of the parameter (the data type is detected automatically) the parameter elements appears as textual / numeric entry field, choice box or as display field for read only parameters. In the screen editor, the parameter element is shown as a sunken rectangle.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.

- **Label**: The label text is drawn above the element.

- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to

higher privilege levels, limiting the group of operators permitted to change the parameter.

- **Use searchable dropdown list**: Checking this mark changes the type of drop down list to be used with a choice parameter. Enumerated values normally are shown as a dropdown list which pops up if you click into the field and can be filtered if you start typing. You always have to select the value with the mouse click or keyboard arrows and then enter key. *If the widget is configured for a height of 40 pixels or more, it appears as a scrollable list rather than as a dropdown*. Checking *Use searchable dropdown lis* also enables the *Allow free text*.

- **Allow free text**: This is also a drop-down list as a searchable drop-down list. In addition, you do not have to select from the dropdown options, but can simply enter any text and save it with the Enter key.

- **Use spin buttons**: Checking this mark enables "spin buttons" for numeric entry fields. Clicking these small arrow symbols at the right edge of the entry field increases/decreases the value by a certain amount. clicking them with the right mouse button increases/decreases by a larger amount.Enabling spin buttons with automatic step size automatically enabled the "Fixed step size".

- **Fixed step size**: Checking this mark permits to set user defined increment values for the spin buttons. If not checked, the spin button increment is derived from the least significant digit shown in the field, the large (right mouse button) increment is ten time that.

    - **Small steps**: The spin button increment value to be applied with left mouse button clicks.
    - **Large steps**: The right mouse button spin button increment. The program assumes 10 times the small steps if the large steps field is left empty.

- **Font**: Font contains two options:

    - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
    - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.

- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.

    - **Enable ID**: This field defines the ID for the enabling parameter.
    - **Enable value**: The value belonging to the parameter above. The actual value of

the parameter

- **Query before applying changes**: Check this mark to make the element show a query before a parameter gets actually commanded.
    - **Query text**: The query text to be shown in the pop-up window. You may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern `$P` get replaced by the parameter name (message ID). Any occurrences of the pattern `$V` get replaced by the new value to set.
- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

NOTE

- The options "use spin buttons" (automatic step size) and "use editable drop down list" internally use the same use the same property of the screen element. Hence, these options are linked in way that if you enable one of these options, the other one is selected as well. This is done to maintain compatibility to older versions of the software. This linkage means no limitation, as a parameter element never can be a numeric entry field and a choice list at the same time.
- If you enlarge the height for a choice parameter widget to more than 40 pixels, the software uses a list widget with scroll bars rather than a drop down selection box.
- If you enlarge the height for a entry field to more than 50 pixels, the software uses a multi line entry field widget. Multi line entry fields are useful only for very few types of parameters, they are not recommended for general use.

2.16.2.1.3 Display element

The display element also known as read-only-parameter element is used to display M&C parameters as read-only. It looks much alike the parameter element, but never allows to

change the parameter it displays.

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Draw frameless with transparent background**: Activate this option so that the element is drawn without an input field frame and with a transparent background. With this option, the read-only parameter element can be used as a label that changes its text with the value of a variable. Please note that you cannot use the *color (Variable background color)* option (described below) together with the frameless / transparent background option.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.
  - **Enable ID**: This field defines the ID for the enabling parameter.
  - **Enable value**: The value belonging to the parameter above. The actual value of the parameter
- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
  - **Color ID**: The message ID of the parameter which controls the background color of this element.
  - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
  - **Value**: The parameter value of the actually selected table entry.

- **Bold**: Draws the text bold if the parameter value matches.
- **Add**: To add a new value / color pair to the list, fill the value and color fields.
- **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
- **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.4 Radio button element

The radio buttons display element is a component for displaying and editing M&C parameters of the CHOICE type in the form of a row of radio buttons. Depending on the height of the element, the radio buttons are arranged in a row or in a column. In the screen editor, the parameter element is displayed as a simple rectangle.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- **Draw frame**: Checking this checkbox makes the element show a rectangular frame around the radio buttons.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.
  - **Enable ID**: This field defines the ID for the enabling parameter.
  - **Enable value**: The value belonging to the parameter above. The actual value of

the parameter

- **Query before applying changes**: Check this mark to make the element show a query before a parameter gets actually commanded.
    - **Query text**: The query text to be shown in the pop-up window. You may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern `$P` get replaced by the parameter name (message ID). Any occurrences of the pattern `$V` get replaced by the new value to set.
- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.5 Text element

The text display element shows a label (a one line text string) with a selectable font / color.

The attributes listed below may be configured to customize a display element of this type:

- **Label**: The label text is drawn above the element.
- **Font**: Font contains two options:
    - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
    - **color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.

2.16.2.1.6 Frame element

The Frame display element draws a sunken 3D frame, which is intended to be used to group other elements. *Beside the position in the window there are no user configurable attributes to set.*

NOTE

- The 3D frame's inside area is transparent, it does not conceal the screen elements it encloses. In the screen editor, however the 3D frame grabs all mouse clicks, so you might not be able to mode the elements which are inside the frame. In this case move the 3D frame to the back by pressing *push element to background* while the frame element is selected. The other elements are now "on top" of the frame element and may be addressed by the mouse independently.

2.16.2.1.7 Rect element

The rect (rectangle) display element draws a rectangle with a selectable color.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Color**: The color to be used to draw the rectangle unless a variable line color is specified. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Filled rectangle**: If this checkbox is checked, a solid rectangle is drawn. The*raw bold* property in the color translation table has no effect when the rectangle is drawn solid.
- **Use variable line color**: Check this checkbox to let draw the rectangle with a color specified by the value of the variable specified by 'message ID'. Unless you check the mark 'use color translation table'
- **Use color translation table**: Checking this checkbox enables a color translation table which works much like the variable background color feature for entry elements. Depending on the actual value of the variable addressed by 'message ID', the element's line color and thickness is set using the translation table shown in the lower left corner of the dialog. If the actual value of the variable does not match any of the table entries or if no message ID is specified, the first color in the list is used to draw the element's background.
    - **Value**: The parameter value of the actually selected table entry.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Bold**: Specifies the line width of the currently selected table entry. If the checkmark is checked, the line is drawn 3 pixels wide, if the checkbox is not checked, the line width is 1 pixel.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.

- **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.8 Arrow element

The arrow display element draws a horizontal or a vertical line and optionally an arrowhead. The line's color is selectable. Strictly speaking the Line / Arrow element is a rectangle with only one or two sides drawn.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Color**: The color to be used to draw the line unless a*use variable line color*is specified. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Arrows and lines**: You can choose the arrow or line from options. Simply click the option.
- **Use variable line color**: Check this checkbox to let draw the line with a color specified by the value of the variable specified by 'message ID'. Unless you check the mark 'use color translation table'.
- **Use color translation table**: Checking this checkbox enables a color translation table which works much like the variable background color feature for entry elements. Depending on the actual value of the variable addressed by 'message ID', the element's line color and thickness is set using the translation table shown in the lower left corner of the dialog. If the actual value of the variable does not match any of the table entries or if no message ID is specified, the first color in the list is used to draw the element's background.
  - **Value**: The parameter value of the actually selected table entry.
  - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
  - **Bold**: Specifies the line width of the currently selected table entry. If the checkmark is checked, the line is drawn 3 pixels wide, if the checkbox is not checked, the line width is 1 pixel.
  - **Add**: To add a new value / color pair to the list, fill the value and color fields.
  - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
  - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.9 Icon element

The icon display element shows an arbitrary GIF/JPEG picture. The image file must reside in the ./images directory on the M&C/NMS server.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Icon**: Selects the name of the image/ icon from dropdown list. Once you selected the icon preview will be displayed below the icon input field.

Using an icon to display a parameter

If you specify a message ID with the element properties of an Icon , the image displayed will change with the parameter value addressed by the message identifier.

You must supply one image file for each value the parameter may have. The parameter value must appear in the file name just in front of the gif/jpg suffix, enclosed in dots. All images used for one parameter should be of the same size. If the Icon element receives a parameter value it does not find an image file for, a black rectangle is displayed instead of an image.

Example: Let's assume you want to visualize a TX ON/OFF switch with an antenna symbol changing it's color with the position of the switch. Supply two images called "transmission.ON.gif" and "transmission.OFF.gif" which symbolize these states as you want. Notice that the ON/OFF keywords in the file names exactly reflect the possible values of this parameter and that the file names only differ in this point. Select one of image files for the icon element and enter the message Id of the TX-ON/OFF parameter into the appropriate field.

The mechanism used to select the image file to display mostly limits the usage of dynamically changing icons to CHOICE parameters. Merely numeric (integer) parameters with a very limited number of possible values come into question, too.

Using an icon to display a device summary fault

With device summary faults ("DEVICENAME.fault"), and subsystem faults ("SUBSYSTEM. ... .fault") the Icon element translates the fault state to "true" for FAULT and "false" for "OK.". This is because the sat-nms software provides the individual fault flags in a device driver as boolean values, the summary fault however is represented by a text which can be one of "OK.", "WARNING" and "Summary FAULT". Translating the OK/FAULT values lets you use the same image files for individual and summary fault states.

To display a device summary fault as an color changing icon, you have to provide three image files, for example:

MyIcon.false.gif for the "OK" state **MyIcon.WARNING.gif** for the "WARNING" state **MyIcon.true.gif** for the "FAULT" state

2.16.2.1.10 Device icon element

The device icon display element represents a device in the M&C user interface. It displays the status of the device by it's color/shape and gives access to the *device details panel* panel for this particular device by a mouse click.

The attributes listed below may be configured to customize a display element of this type:

- **Device name**: The name of the device this element stands for.
- **Icon**: Selects the device icon / image from dropdown list to display. Once you selected the icon preview will be displayed below the icon input field. The sat-nms software at least provides three predefined icons for this purpose:
  - : device-rectangle.gif
  - : device-square.gif
  - : device-minirect.gif
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- **Show device name**: Check this checkbox to display the name of the device as a label in the icon. Unchecked can be useful if very small icons are used in which the label would not fit, or if a label is to be used that is different from the name of the device.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Signal if device actually is redundant**: Check this checkbox to apply a special coloring for devices which are in the redundant signal path of a redundancy switching facility.
  - **Redundancy signal ID**: The message ID of a parameter which gives the information if the device actually operates as the redundant one or if it is operational. For a 1:1 TWTA redundancy this could be the position parameter of the waveguide switch which selects the signal from the two TWTAs
  - **Redundancy signal value**: If the value of the parameter addressed by above message ID matches the value given in this field, the device icon is shown as "redundant" (depends in the icon set used, the factory icon sets use a blue background color for this).

Using customized images

You may supply your own images for the device icon display element. To do so, place seven images into the *./images* directory named "xxxxx.gif", "xxxxx-F.gif" (fault), "xxxxx-C.gif" (communication fault), "xxxxx-S.gif (fault suppressed)", "xxxxx-R.gif" (redundant), "xxxxx-O.gif" (out of service) and "xxxxx-M.gif" (maintenance). You should consider the following:

- All images should be of the same size.
- The software places the device name at the lower left corner of the image using the selected font and color. All four images should use appropriate colors in this area to keep the device name visible.
- Beside GIF-images the software also accepts PNG images (file extension .png) and JPG images (file extension .jpg). All images of a set must be of the same type (GIF/PNG/JPG).

Using a customized 'transmitting' emblem

For devices which provide the 'info.signal.on' parameter to report that the device is actually transmitting or receiving, the device icon shows an emblem at it's upper right corner. The software uses the 'signal-on-emblem.png' image as a default for this.

You may define individual emblems for each image set by placing a "xxxxx-X.gif" (or .png/.jpg) image in the ./images folder with xxxxx being the base name of the device icon. The software will use this file for the transmitting emblem of this particular icon set. The file extension of the emblem must match the extension of the base icon image.

2.16.2.1.11 Parameter button element

The Parameter Button display element is a button which sends a parameter value if pressed. A frequently used application for the parameter button is a RF-OFF button which sends a "tx.on=OFF" to a certain device. Beside this, a parameter button also may be programmed to play a parameter setting macro.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- **Start macro**: Selects if the button shall start a macro (checked) or send a certain parameter (unchecked).
- **Use icon instead of label**: Check this checkbox in order to make the button show an icon instead of a label text.
    - **Icon**: Provides a list of all available icons / images on the sat-nms server.
- **Specify values to send**: Checking this checkbox let you define the parameter value to be sent with a button click. Without this option the button sends the label text as a parameter value:

    - **Parameter value**: The value to be sent when the button is clicked.

- **Font**: Font contains two options:

    - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
    - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.

- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.

    - **Enable ID**: This field defines the ID for the enabling parameter.
    - **Enable value**: The value belonging to the parameter above. The actual value of the parameter

- **Query before applying changes**: Check this mark to make the element show a query before a parameter gets actually commanded.

    - **Query text**: The query text to be shown in the pop-up window. You may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern `$P` get replaced by the parameter name (message ID). Any occurrences of the pattern `$V` get replaced by the new value to set.

- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.

    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then

click delete.

2.16.2.1.12 Latching button element

The Latching Button display element works much like the Parameter Button element described in the previous chapter, but specialized to show and control an enumeration parameter which knows exactly two states (e.g. on/off or true/false).

When the operator changes the state of the button by clicking it once, the latching button sends the 'other' parameter value to the device. On the other hand, if some other instance in the system changes the parameter state, the latching button recognizes this and changes the up/down state of the button accordingly.

Like the parameter button, the latching button may be labeled with text or an image. To reflect the actual state, the latching button always is configured with two text string or two image names which are shown according to the actual parameter value.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label up**: This label text is drawn on the button in "up" state.
- **Label down**: This label text is drawn on the button in "down" state..
- **Value up**: The value to be sent when the button is clicked an changes to "up" state.
- **Value down**: The value to be sent when the button is clicked an changes to "down" state.
- **Privilege level**: The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- **Use icon instead of label**: Check this checkbox in order to make the button show an icon instead of a label text.
    - **Icon up**: The name of the icon to be shown in "up" state of the button. The field provides a list of all available icons / images on the sat-nms server.
    - **Icon down**: he name of the icon to be shown in "down" state of the button. The field provides a list of all available icons / images on the sat-nms server.
- **Font**: Font contains two options:
    - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except

the smallest one. In the latter case the label is drawn using this small font size, too.
    - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.

- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.

    - **Enable ID**: This field defines the ID for the enabling parameter.
    - **Enable value**: The value belonging to the parameter above. The actual value of the parameter

- **Query before applying changes**: Check this mark to make the element show a query before a parameter gets actually commanded.

    - **Query text**: The query text to be shown in the pop-up window. You may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern `$P` get replaced by the parameter name (message ID). Any occurrences of the pattern `$V` get replaced by the new value to set.

- **Color**: Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by *color id*, the element's background is set using the translation table. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.

    - **Color ID**: The message ID of the parameter which controls the background color of this element.
    - **Color**: The color specification of the actually selected table entry. Clicking to this opens a color chooser dialog which lets you select/compose the the color to your needs.
    - **Value**: The parameter value of the actually selected table entry.
    - **Bold**: Draws the text bold if the parameter value matches.
    - **Add**: To add a new value / color pair to the list, fill the value and color fields.
    - **Set**: To change the color assigned to a given parameter value, first select the table entry of interest, change the color value.
    - **Delete**: To delete a value / color pair from the list, first select the table entry then click delete.

2.16.2.1.13 Gauge element

The Gauge display element shows a numeric parameter value as a horizontal bar in an entry field like frame. The gauge element is capable to adjust the scale factor for the gauge

automatically from the parameter's range definition. Alternatively the scale parameters may set explicitly.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Specify range**: Check this checkbox to define the range which is displayed by the gauge element.
  - **min value**: The minimum value of the displayed range.
  - **max value**: The maximum value of the displayed range.
- **Specify thresholds**: Check this checkbox to define the threshold values which are used for RGR and GYR color schemes.
  - **min thresholds**: The minimum value of the threshold range.
  - **max thresholds**: The maximum value of the threshold range.
- **Graph color**: The gauge element knows three color schemes, allowing to change the bar color dynamically with the monitored value. *Please note, unless you specify min/max thresholds for the monitored value, only the fixed color mode is available.*
  - **Fixed**: The bar is displayed with the color defined with in the above color filed.
  - **R-G-R (red, green, red)**: The bar is displayed green while the value is between the minimum and maximum thresholds. If the value is outside the limits, the bar is displayed red.
  - **R-Y-G (red, yellow, green)**: The bar is displayed red while the value is below the minimum threshold value. It changes to yellow if the value is between the minimum and maximum thresholds. If the value exceeds the maximum threshold, the bar color changes to green.
  - **G-Y-R (green, yellow, red)**: The bar is displayed green while the value is below the minimum threshold value. It changes to yellow if the value is between the minimum and maximum thresholds. If the value exceeds the maximum threshold, the bar color changes to red.
- **show ticks**: check this to make the element show scale ticks vor the min/max and threshold values below the gauge.
- **show ticks with labels**: With this mark checked, below the gauge element labeled ticks are shown for the min/max and threshold values. (requires "show ticks"). Bar graph elements with this option enabled should be configured with a width that is sufficient to

show the scale labels without overlap.

2.16.2.1.14 Chart element

The (strip) Chart display element shows a strip chart of a numeric parameter. The chart advances with a constant speed of 1 pixel / second. The default y-scale is 1/division but may be changed by clicking to the menu button in the chart on top right corner.

By default, the strip chart element lets the y-scale offset follow the displayed value that the recent measurement samples are shown in the diagram. This behavior is optimized for applications where the strip chart shall indicate a 'trend' for the displayed value, using an element height of only 50 pixels or less.

Optionally you may disable this variable scaling by defining fixed scale values in the 'Text / Image Name' field. With this fixed y-axis scaling an additional threshold check may be added, turning the chart background to red if the actual value exceeds the defined limits. When used with limit checking, the height of the chart should be at least 100 pixels.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.
- **Label**: The label text is drawn above the element.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **Color**: The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Mode**: Selects the display mode of the chart element. Four modes are available:
  - **Auto scale**: This is the standard display mode. The strip chart aligns the reference value (the medium display line) in a way that every new display point is shown on this line. The initial scale is 1 unit/div.
  - **Fixed scale**: Works like the auto scale mode, but the scale value gets initialized with a user defined value.
    - **Scale**: The initial scale value used with the 'floating, fixed scale' mode.
  - **Fixed range**: The min/max values to display are defined explicitly, there is no

automatic reference level alignment with this mode.

- **min value**: The minimum value to display for the fixed range display modes.
- **max value**: The maximum value to display for the fixed range display modes.

- **Fixed thresholds**: It is like the above mode, but additionally there are min/max thresholds defined. If the actual value of the monitored parameter, the display background changes to red.

  - **min value**: The minimum value to display for the fixed range display modes.
  - **max value**: The maximum value to display for the fixed range display modes.
  - **min threshold**: The lower threshold value for the "fixed range & thresholds" mode.
  - **max threshold**: The upper threshold value for the "fixed range & thresholds" mode.

2.16.2.1.15 Spectrum element

The Spectrum Display element embeds the spectrum display of a spectrum analyzer device in the screen. The display may be scaled to an almost arbitrary size, however the following limitations apply:

- The annotation labels which reflect the actual spectrum analyzer settings are not scaled with the display element. If the element is very small, the labels may overlap and become unreadable.
- Resizing the spectrum display element is only possible in certain steps, the grid behind the spectrum curve is always made up of 10*n + 11 pixels in one dimension. The screen editor software knows about this, snaps the size of the element to the next (smaller) valid size after you release the mouse button.

The attributes listed below may be configured to customize a display element of this type:

- **Device name**: The name of the spectrum analyzer device with MNC name (MNC.devicename) to operate.
- **Label**: The label text is drawn above the element.
- **Privilege level**: The privilege level necessary to operate the spectrum analyzer device.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
  - **Color**: The color attribute of a element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.

2.16.2.1.16 Target list element

The Target List screen element lets you embed the list of targets of a SatService-ACU-ODM

antenna controller in the screen. Targets may be recalled (which moves the antenna to the stored position and sets the tracking parameters associated with this target), saved or deleted.

Please note, that the ODM Target List screen element only works with SatService-ACU-ODM antenna controller device type, other antenna controllers are not supported.

The attributes listed below may be configured to customize a display element of this type:

- **Device name**: This parameter is interpreted differently depending on the context where the ODM Target List screen element resides:
  - When placed in a user defined screen or in the main screen of the application, 'device name' must be set to the name of the ODM device it shall refer to.
  - When placed in the device screen of another device which defines a configuration variable with the antenna controller device name, the name of this configuration variable must be entered as the 'device name'. Finally, when used in the device screen of the ODM device itself, 'device name' must de set to '@'
- **Privilege level**: The privilege level necessary to operate the target list device.
- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.
  - **Enable ID**: This field defines the ID for the enabling parameter.
  - **Enable value**: The value belonging to the parameter above. The actual value of the parameter

2.16.2.1.17 XY chart element

This element shows the relation of two numeric variables in an X/Y diagram, featuring a 'track' which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

The X/Y chart display may be scaled to an almost arbitrary size, however the following limitations apply:

- The annotation labels which reflect the actual display settings are not scaled with the display element. If the element is very small, the labels may overlap and become unreadable.
- Resizing the X/Y chart display element is only possible in certain steps, the grid behind the curve is always made up of a number pixels in one dimension which is divisible by the configured number of diagram divisions without remainder. The screen editor software knows about this, snaps the size of the element to the next (smaller) valid size after you release the mouse button.

The attributes listed below may be configured to customize a display element of this type:

- **MessageID(x) message ID (y)**: The message IDs of the parameters to show.
- **Label**: The annotation label displayed above the X/Y chart.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the text shown in the parameter field itself. The

label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
    - **Color**: The color of the label displayed above the X/Y chart. Colors may be specified by using a hexadecimal RGB notation. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- **Chart settings**
    - **Divisions**: The number of divisions shown in the diagram for both directions. You may enter arbitrary values here, but to achieve some readability, you are encouraged to us common values like 2 ('hair cross'), 4 or 10.
    - **Interval**: The update time interval for the display in seconds. 0.1 means to add every 100msec a new value to the display and remove the oldest value from the buffer at the same time.
    - **Buffer size**: The display maintains a "first in first out" buffer of a size defined with this parameter. The buffer provides a short time memory the display shows as a trace of past values. Values in the range 100-300 are a good choice, larger values may slow down the client application. Please note, that the buffer gets cleared when the window is closed.
- **Origin and scale**
    - **x origin y origin**: The origin values refer to the center of the diagram. Using an even number of divisions helps to identify the diagram center as the cross point of the middle grid lines
    - **x scale y scale**: The scale values are per division. with x origin = 0.0 and x scale = 1.0 an x value of 1.0 gets displayed one division right of the diagram center.
- **Monitor limits**: With this checked, the display monitors the actual X/Y values to be within the limits defined below. The limit values are shown as a dark red rectangle. When the while the actual X/Y values exceed the limits, the diagram background becomes red.
    - **min x value, max x value, min y value and max y value**: The limit values for X and Y to be checked.

2.16.2.1.18 Thumbnail icon element

The Thumbnail Icon element lets you display a thumbnail of the video actually processed by a device like a an encoder, decoder or gateway. The displayed icon is a static image, updated every couple of seconds.

The attributes listed below may be configured to customize a display element of this type:

- **Image URL**: The URL where to fetch the image.
- **Reload time**: The interval when the image shall be reloaded (seconds)
- **URL type**: One of PLAIN or ADVANCED. With PLAIN the URL is used as it is, with ADVANCED CGI parameters for the actual time and the image width and height are appended to the URL.

REMARKS

- The device to get the thumbnail images from must support this feature.
- The device to get the thumbnail images from must be accessible in the network from the client's point of view.

- Thumbnail Icons are - when resized by mouse drag in the GUI editor - forced to 16:9 ratio. If another image ratio is required, image width and height must be entered in the dialog shown above. in PLAIN mode, the image received from the device gets scaled to fit the screen element size, in ADVANCED mode no scaling is done as the software queries the image already in the correct size from the device.

2.16.2.1.19 Switch element

The Switch Icon display element is a special version of the Device Icon element which may be used to visualize the position of a switch in a user interface screen designed as a block diagram. The switch icon has all capabilities of a plain device icon display element. The context menu shown with this right mouse button additionally contains an option to toggle the switch position. The switch icon element -- unlike the plain device icon element -- does not support user defined images.

The attributes listed below may be configured to customize a display element of this type:

- **Device name**: The name of the device this element stands for. You also may enter the complete ID of the parameter which controls the switch position for devices which manage more than one switch or use a non-standard switch position parameter name.
- **Icon**: Select the shape of the icon from the given switch symbols or a user defined switch. .ts user defined switchThe (base-) name of a set of userdefined switch icons. See below how to use your own switch icons.
- **Privilege level**: The privilege level necessary to toggle the switch position. To change the device mode (OPERATIONAL, OUT-OF-SERVICE), at least a privilege of 100 is necessary, or the value set here if this is higher.
- **Touch screen mode**: Selecting this option makes the switch icon behave differently, optimized for touch screen usage. You may toggle the switch position by touching the icon twice. This first time you touch it, the icon shows a turquoise ball in the upper left corner. This ball disappears after 2 seconds. If you touch the icon a second time before the turquoise ball disappears, the switch is actuated. There is no right mouse button context menu in this mode, no way to go into the device screen or set the device OUT-OF-SERVICE.
- **Font**: Font contains two options:
  - **Font**: The font attribute applies to the device name, too.
  - **Color**: The color attribute of a parameter element sets the color of the device name shown with the icon.
- **Conditionally enable widget**: Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this to activate this feature.
  - **Enable ID**: This field defines the ID for the enabling parameter.
  - **Enable value**: The value belonging to the parameter above. The actual value of the parameter
- **Query before toggling switch**: Check this mark to make the element show a query before a parameter gets actually commanded.
  - **Query text**: The query text to be shown in the pop-up window. You may leave this

field empty, the program uses a standard query text in this case. In the question text, placeholders may used for two values: Any occurrences of the pattern `$P` get replaced by the parameter name (message ID). Any occurrences of the pattern `$V` get replaced by the new value to set.

Remarks

Configuring a confirmation question as described above, only affects the 'Toggle position' function provided by the switch icon's right mouse context menu. Setting the switch position from the device dialog always works without a confirmation request.

User defined switch icons

The sat-nms software comes with 6 predefined switch icons. You may add your own icons if none of the predefined switches matches your needs.

For a user defined switch you have to create a set of ten PNG image files / icons. Each icon shows one combination of switch position and device state. All icons of a set must be of the same size, the files must be located in the *images* subdirectory of the sat-nms server installation and they must follow the naming scheme described below.

The file name of a switch icon consists of the user defined name, followed by a dash and a 2-character status code. All switch icons must be PNG images and end with the file suffix ".png" (lower case). The first character of the status code reflects the switch position, the second one the device's fault/operation state. The following codes are defined:

pos.	character	description
1	A	switch is in position A (or OFF)
1	B	switch is in position B (or ON)
2	N	switch device is in normal operation, no fault is pending
2	W	switch device shows a warning
2	F	switch device shows a fault
2	S	switch device is set FAULT-SUPPRESSED
2	O	switch device is set OUT-OF-SERVICE

Combining the two switch positions with five device status codes results in totally 10 icons which show all possible states of a switch. If you create a switch named "myswitch", the files would be:

myswitch-AN.png myswitch-AW.png myswitch-AF.png myswitch-AS.png myswitch-AO.png myswitch-BN.png myswitch-BW.png myswitch-BF.png myswitch-BS.png myswitch-BO.png

Replacing the predefined icons

In in your sat-nms installation mostly user defined switch symbols are used, you may replace some or all of the predefined switch symbols offered by the Switch Icon dialog with your favorite icons. To configure this, edit the file "client.properties" in the base directory of the sat-nms server installation and add some or all lines of the following:

gui.editor.predefinedSwitch.1=myswitch1 (replace myswitch1 by the name of the icon set to use) **gui.editor.predefinedSwitch.2=myswitch2 gui.editor.predefinedSwitch.3=myswitch3 gui.editor.predefinedSwitch.4=myswitch4 gui.editor.predefinedSwitch.5=myswitch5 gui.editor.predefinedSwitch.6=myswitch6**

Please note, this does not replace / delete the factory icons. Screens which make use of the factory switches will still use them as before and you are still able to edit screens with factory switch icons by explicitly define their names in the *user defined switch* field. The names of the factory switch icons are *wgswitch* , *wgswitch-mirrored* , *cxswitch* , *cxswitch-mirrored* , *lswitch* and *rswitch* .

User defined switch icons may be of any size, even larger than the factory icons. When used as predefined switch icons, such large icons will show up in the dialog with their upper left corner only. In the screen the icons gets displayed with it's full size.

2.16.2.1.20 AzEl element

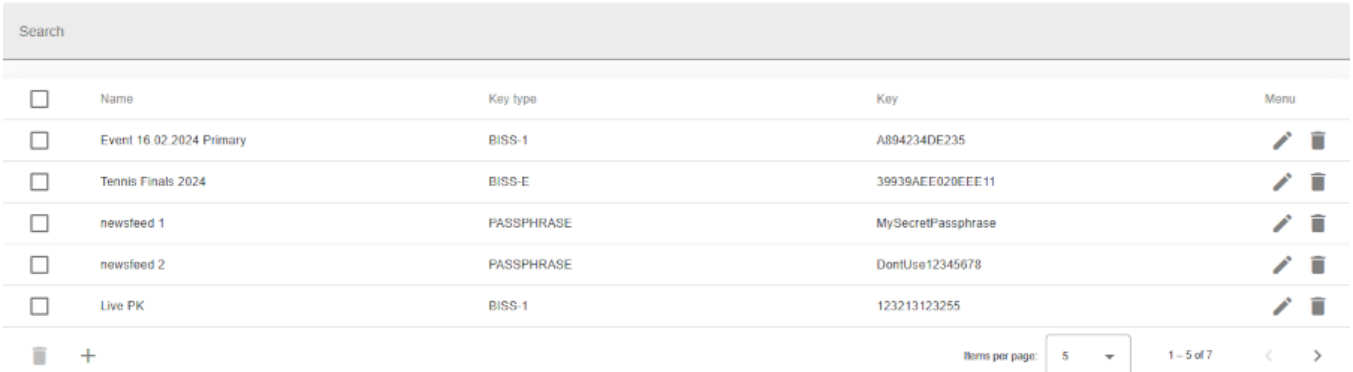
The AzElElement shows the tracking history of an antenna controller as a cloud of points in a elevation over azimuth coordinate system. Actually only SatService antenna controllers provide the tracking history data for this screen element.

The attributes listed below may be configured to customize a display element of this type:

- **Message ID**: The message ID addresses the parameter the element shall display or edit. The message ID consists of the MNC name, device name followed by a colon and the name of the parameter.ANT.11.XMIT-1.tx.on for example addresses the variable tx.on at a device named XMIT-1 at MNC.

2.17 Stream keys

The Stream key is record holds the information about one BISS or SRT key in the database. The Table provides a detailed interface to manage keys. It allows users to view, edit, delete, and create key efficiently.



Search bar

- Located at the top of the table.
- Allows users to filter keys by keywords, such as name, key type and key.

Header row

- You can sort rows in ascending or descending order by clicking on the column header (e.g. name, key, etc.).
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
  - **Location**: The checkbox column is located at the far left of the table.
  - **Purpose**: Allows users to select one or multiple keys for delete key(s).
  - **Functionality**:
    - **Single Selection**: Click on a single checkbox to select an individual row.
    - **Multiple Selection**: Click multiple checkboxes to select multiple rows at once.
    - **Select All**: The checkbox in the header row allows you to select or deselect all rows on all pages.
    - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **Name**: The user defined unique name of stream key.
- **Key type**: The key type, one of 'BISS-1', 'BISS-E' or 'PASSPHRASE'.
- **Key**: The stored key in database.
- **Functions**: Action icons for managing the key:
  - **Edit**: Pencil icon to modify/edit the key.
  - **Delete**: Trash can icon to delete the key.

Bottom toolbar

- **Delete selected keys**: A button to delete selected keys/rows from checkbox column.
- **Create key**: + button is used to create new stream key.
- **Pagination controls**: Allows users to set the number of items displayed per page and

navigate between pages.

2.17.1 Features

Create key

- Click the + button at the bottom of the table to add a new event.
- Input the required details.

Edit key

- Click the *pencil icon* in the Functions column to modify key details.
- Adjust the settings and save.

Delete event

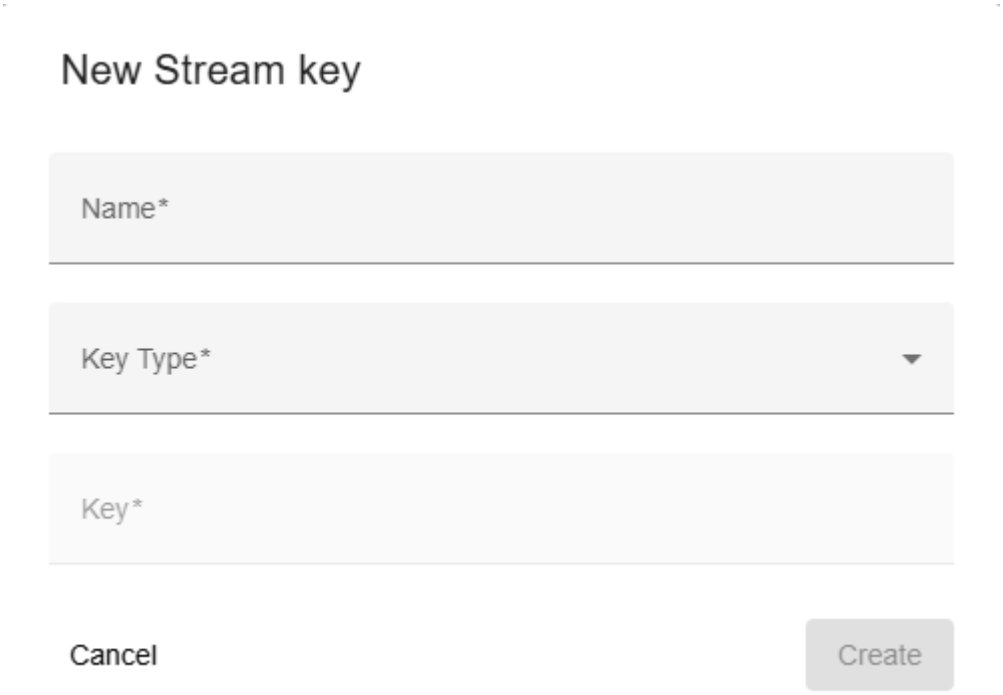
- Click the *trash icon* in the Functions column to delete an key.
- Confirm the action in the pop-up dialog to delete the key.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.17.2 Dialogs

Stream key add/edit dialog



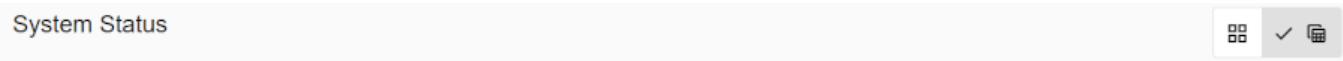
- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new key or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new stream key, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Name**: The user defined name of this stream key. This name must be unique, as the operator selects the key according to this name.
  - **Key type**: The key type, one of *BISS-1*, *BISS-E* or *PASSPHRASE*.
  - **Key**: The key value associated with the id and name in this record. For *biss* keys this is a string entirely consisting of the hexadecimal characters 0123456789ABCDEF and it is either 12 or 16 characters long. For other key types, the key is a free text.
- **Actions**:
  - **Create/Save**:
    - For new stream key, this creates the event with the provided details.
    - For existing keys, this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete kex confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column or trash icon from bottom toolbar .
- **Content**: Asks the user to confirm whether they want to delete the selected event(s).
- **Actions**:
  - **Yes**: Deletes the event(s) permanently.
  - **Cancel**: Cancels the deletion action.

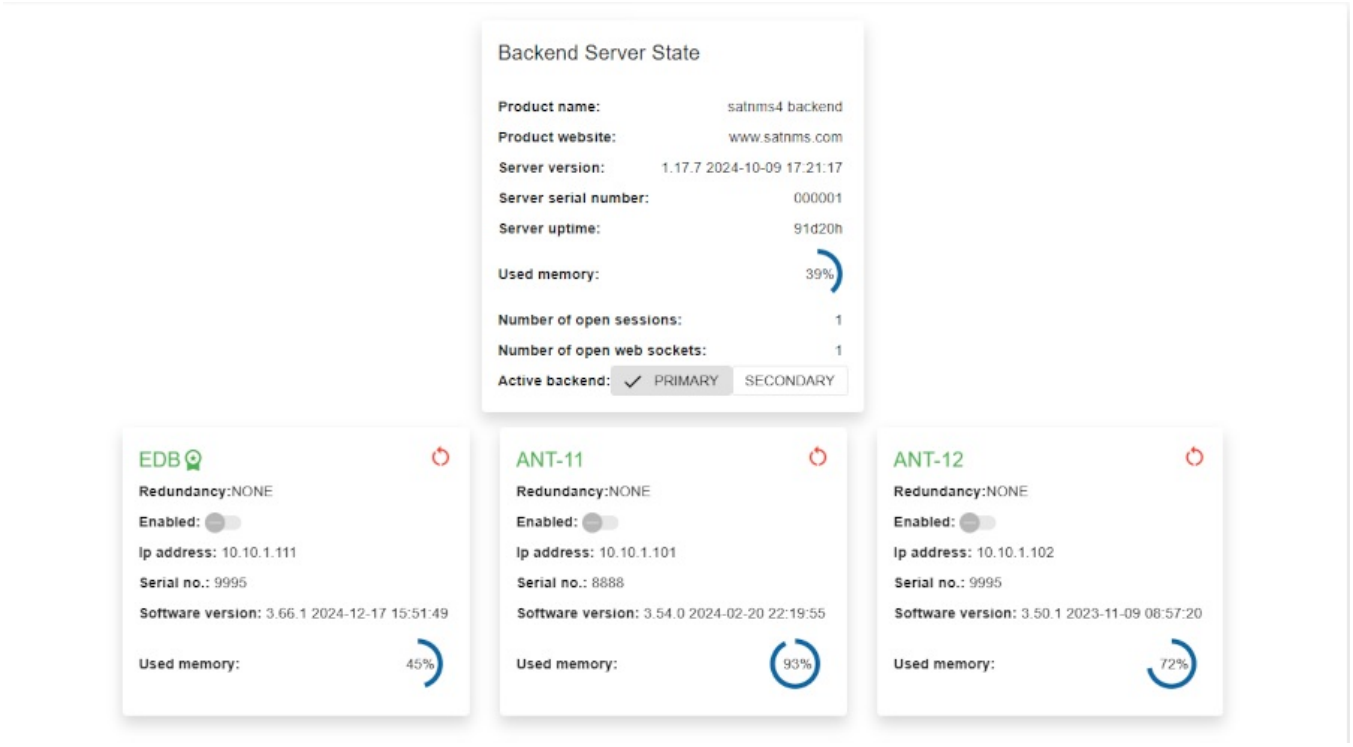
2.18 System status

This page gives you status information about the backend server.

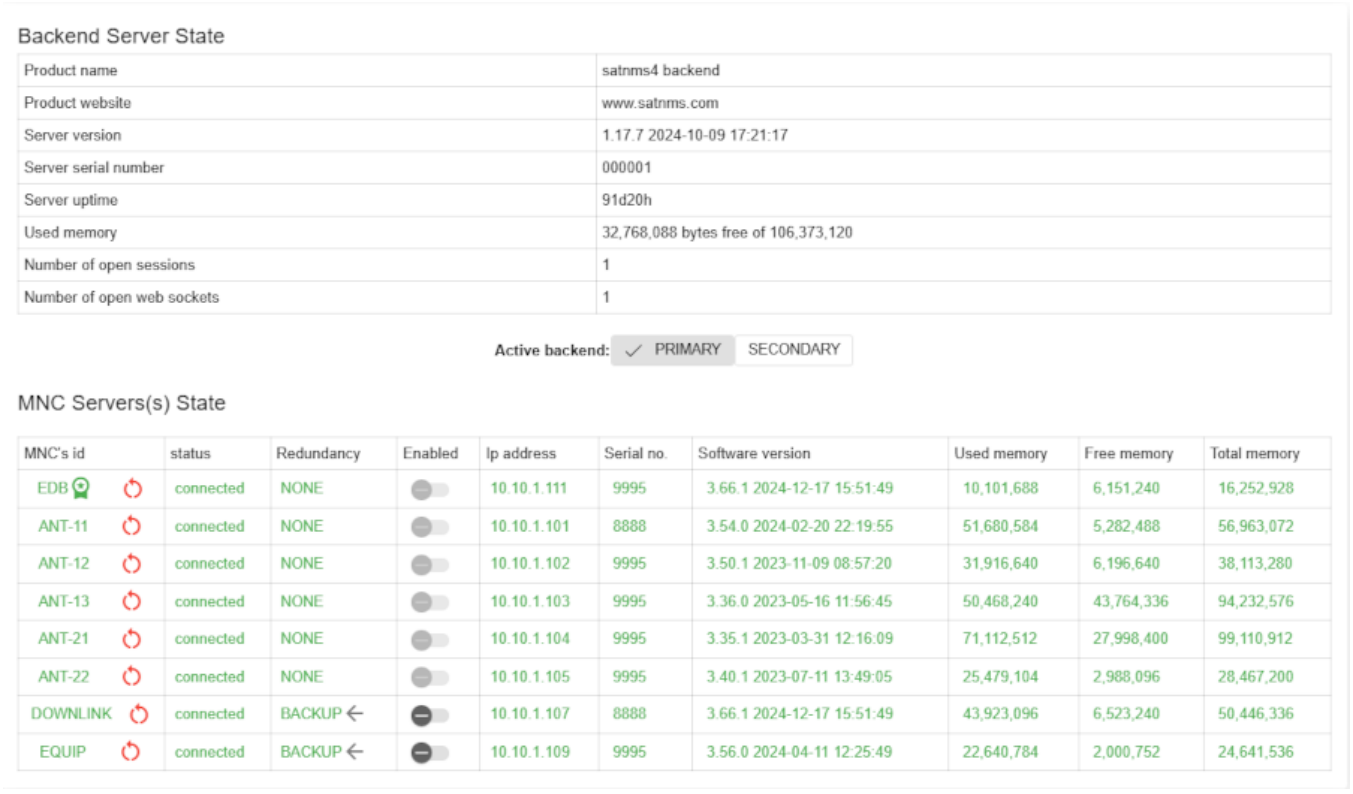


1. **Title**: Title of page.
2. **View options**: At the top right of the page are view options.
   - Grid view (default)
   - Table view.

2.18.1 Grid view



2.18.2 Table view



Backend Server State

Product name	satnms4 backend
Product website	www.satnms.com
Server version	1.17.7 2024-10-09 17:21:17
Server serial number	000001
Server uptime	91d20h
Used memory	32,768,088 bytes free of 106,373,120
Number of open sessions	1
Number of open web sockets	1

Active backend: PRIMARY SECONDARY

MNC Servers(s) State

MNC's id	status	Redundancy	Enabled	Ip address	Serial no.	Software version	Used memory	Free memory	Total memory
EDB	connected	NONE		10.10.1.111	9995	3.66.1 2024-12-17 15:51:49	10,101,688	6,151,240	16,252,928
ANT-11	connected	NONE		10.10.1.101	8888	3.54.0 2024-02-20 22:19:55	51,680,584	5,282,488	56,963,072
ANT-12	connected	NONE		10.10.1.102	9995	3.50.1 2023-11-09 08:57:20	31,916,640	6,196,640	38,113,280
ANT-13	connected	NONE		10.10.1.103	9995	3.36.0 2023-05-16 11:56:45	50,468,240	43,764,336	94,232,576
ANT-21	connected	NONE		10.10.1.104	9995	3.35.1 2023-03-31 12:16:09	71,112,512	27,998,400	99,110,912
ANT-22	connected	NONE		10.10.1.105	9995	3.40.1 2023-07-11 13:49:05	25,479,104	2,988,096	28,467,200
DOWNLINK	connected	BACKUP ←		10.10.1.107	8888	3.66.1 2024-12-17 15:51:49	43,923,096	6,523,240	50,446,336
EQUIP	connected	BACKUP ←		10.10.1.109	9995	3.56.0 2024-04-11 12:25:49	22,640,784	2,000,752	24,641,536

2.18.3 Common functions and details

Backend server state

- **Product name**: The name of product.
- **Product website**: The SatService website URL.
- **Server version**: The software version actually running.
- **Server serial no.**: The software serial number.
- **Server uptime**: The server uptime.
- **Used memory**: The memory usage of the backend.
- **Number of open sessions**: The number of open sessions.
- **Number of open web sockets**: The number of open (message) websocket connections.
- **Active backend**:
  - The name of the backend which actively controls M&C redundancy switching in a redundant backend configuration.
  - **Functionality**: Clicking on name will change the active backend.

MNC information

Below you will find information about MNC.

- **MNC name**:
  - The name of the M&C.
  - If there is a medal icon that means that MNC is an primary MNC (e.g. EDB)
  - **Restart MNC**: On the right hand side you can restart an MNC by clicking on red reload button.
- **Redundancy**:
  - Shows the state of the redundancy control for this M&C.
  - **Functionality**: Switch from main to backup or backup to main
- **Enabled**: Flag if this redundancy is enabled or not.
- **Ip address**: The IP address of the M&C server as defined in the backend configuration file.
- **Serial no.**: The software serial number.
- **Server version**: The software version actually running.
- **Used memory**:
  - The memory used by MNC.
  - On hover it shows total-, used- and free-memory.

2.18.4 Features

Restart MNC

- Clicking on the red reload button (grid or table view) restarts the MNC.

Active backend

- If you click on the backend name below or near the active backend text, the active backend name is changed.

Switch to main/backup

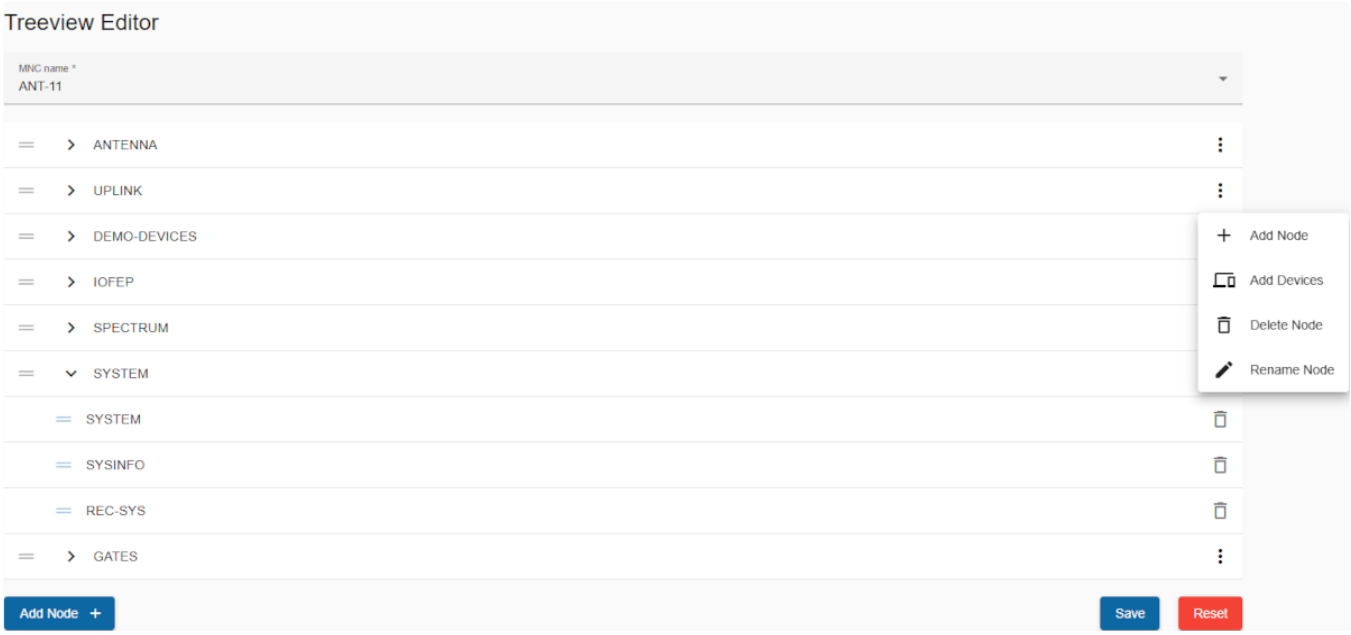
- Click on the arrow buttons (left or right) next to *Redundancy* to change MNC from *Backup to Main or Main to Backup*.

Toggle redundancy

- You can activate or deactivate redundancy using the slider under *Enabled*.

2.19 Treeview editor

The Treeview editor lets you define or modify the tree / subsystem hierarchy used by the M&C Tree View.



Title: Title of page.

MNC name: Below the title, you must select an option from the drop-down list to start editing.

Tree structure

- **Subsystem/Parent node**: are displayed as the top-level entries in the tree. Each parent node can have child nodes nested underneath. Subsystem/Parent nodes can be expanded (▼) or collapsed (►) to hide or reveal their child nodes.
    - **Drag handel**: Appears to the left of the subsystem/parent node name, allowing the node to be reordered or moved into another subsystem/parent node.
    - **Node name**: Displays the name of the subsystem/parent node.
    - **Context-menu (⋮)**: Located to the right of the node name, this menu provides the following options:
        - **Add Node**: Create a new subsystem node under the selected node.
        - **Add Devices**: Attach specific devices to the selected node.
        - **Rename Node**: Change the name of the selected subsystem/parent node.
        - **Delete Node**: Remove the subsystem/parent node and all its children.

- **Device/Child node**: are displayed indented beneath their respective subsystems or parent nodes and cannot exist at the top level (as subsystem). They must always belong to a subsystem.
    - **Drag handel**: Appears to the left of the device/child node name, allowing the node to be reordered within the same subsystem/ parent or moved into another subsystem/parent node.
    - **Node name**: Displays the name of the device/child node.
    - **Delete button**: A trash bin icon on the right allows for quick deletion of the device/child node.
- **Drag and Drop**: Subsystem/Parent nodes and devices/child nodes can be dragged and dropped using the drag handles:
    - **Reordering**: Rearrange nodes within the same subsystem/parent.
    - **Moving**: Move nodes between different subsystems/parents.
    - **Error**: If something is not correct, the snack bar (error) would be displayed with the description of the error at bottom of the page.

Bottom toolbar actions

The Save and reset button is deactivated if no changes have been made.

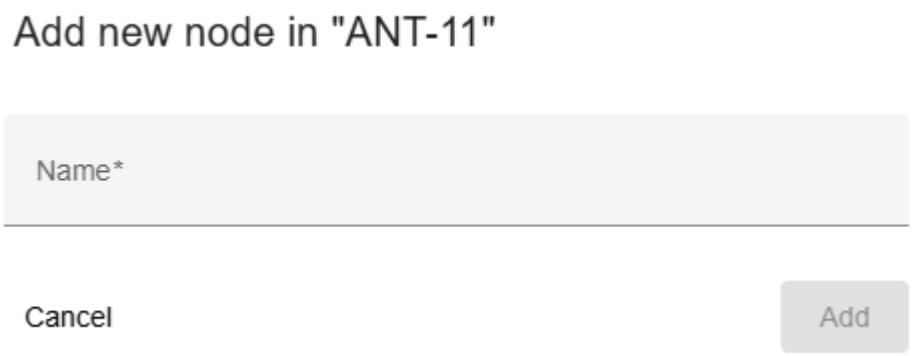
- **Add node**: Use the *Add Node +* button at the bottom of the editor opens a dialog to quickly add a new subsystem/parent node to the tree.
- **Save**: Saves all changes made in the editor. If you have not saved the changes and accidentally leave the editor, a confirmation dialog will appear in which you must give your consent to leave the editor/page.
- **Reset**: Revert the tree structure to its original state.

2.19.1 Functionality

1. **Drag-and-Drop**: Users can drag and drop rows to rearrange the hierarchy dynamically.
2. **Reset changes**: If you made a mistake and want to have old state just click on reset changes to start again from first state.
3. **Unsaved changes**: If you have not saved the changes and accidentally leave the editor, a confirmation dialog will appear in which you must give your consent to leave the editor/page.

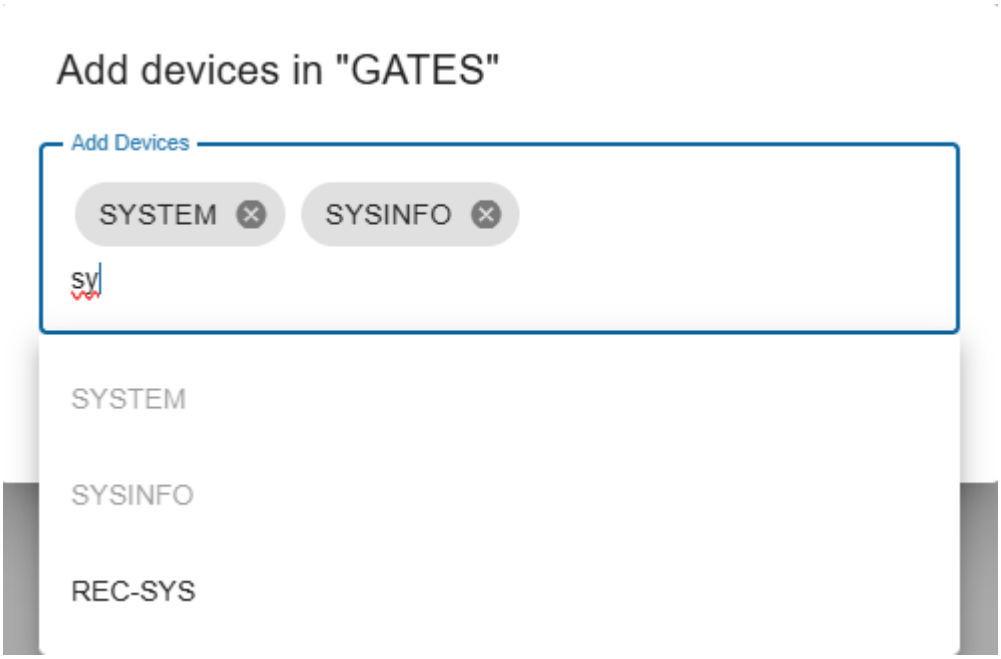
2.19.2 Dialogs

Add/rename node/subsystem



- **Trigger**:
  - when you click on the "Add node +" button on the bottom to create new subsystem/parent node or
  - an on *pencil icon* to rename node from context menu
  - The dialog title clearly indicates whether you are creating a new node on top level or under other node, renaming an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Name**: The name of the subsystem or node. It must be unique compared to other subsystem/node names.
- **Actions**:
  - **Add/Rename**:
    - For new node, this creates the node with the provided name and add devices dialog will be open.
    - For existing node, this renames subsystem/node.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

Add devices



- **Trigger**:
  - when you click on the "Add devices" button from context menu of selected subsystem/parent node.
  - The title of the dialog clearly indicates in which subsystem/parent node you are adding devices to provide clarity about the action being performed.
- **Fields**:
  - **Add devices**:
    - **Device filtering**: The input field includes a dropdown that shows a list of available devices. As the user types in the input field, the list is automatically filtered to match the entered text. Once you have completed the device selection, simply click outside the dialog to hide the dropdown.
    - **Adding Devices**: When a device is selected from the dropdown, it appears as a small tag (chip) displayed above the input field. Users can continue selecting multiple devices, with each device added as a separate tag. Selected device(s) are disabled for the clarity. Each tag includes a small *close (×) icon*. Clicking on the close icon removes the corresponding device from the selection.
- **Actions**:
  - **Add**: Add device(s) to the selected subsystem/parent node.
  - **Cancel**: Closes the dialog without saving.

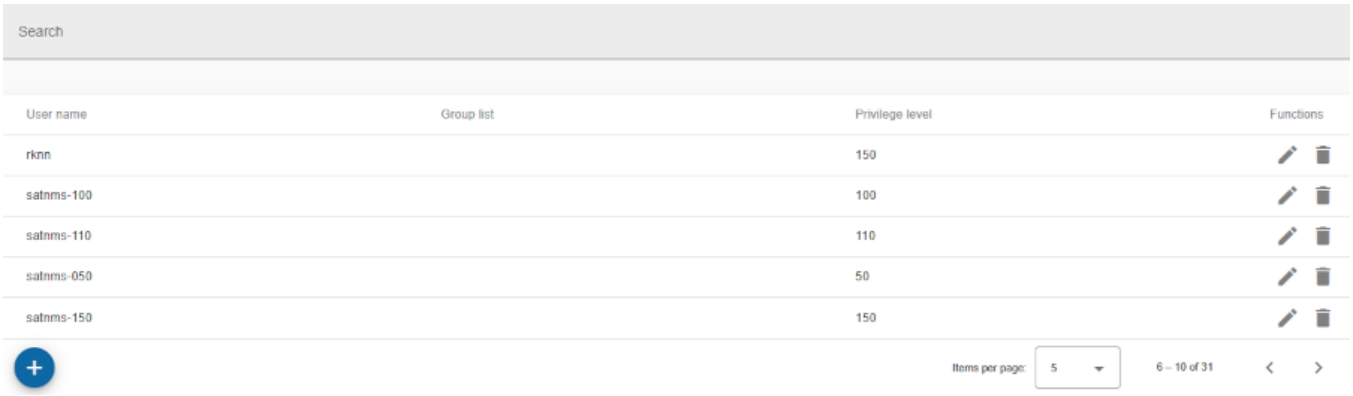
Confirmation dialog

- **Trigger**: Opens when you click the
  - delete node from context menu
  - Trash icon from device/child node.
  - Reset button
  - Leaving the page if there are unsaved changes.
- **Content**: Asks the user to confirm the action, depending on where you trigger the dialog.

- **Actions**:
  - **Yes**: Confirm the action.
  - **Cancel**: Cancels the action.

2.20 User management

The user management lets you manage the list of users which are allowed to operate the software. You need to be a administrator with privilege level 150 or higher to access the user management page.



Search bar

- Located at the top of the table.
- Allows users to filter keys by keywords, such as name, key type and key.

Header row

- You can sort rows in ascending or descending order by clicking on the column header.
- Sorting is applied to the entire dataset, not just the current page.
- The active sort order (ascending or descending) is typically indicated by an arrow or icon near the column name.

Table columns

- **Checkbox column**:
  - **Location**: The checkbox column is located at the far left of the table.
  - **Purpose**: Allows users to select one or multiple rows for delete row(s).
  - **Functionality**:
    - **Single Selection**: Click on a single checkbox to select an individual row.
    - **Multiple Selection**: Click multiple checkboxes to select multiple rows at once.
    - **Select all**: The checkbox in the header row allows you to select or deselect all rows on all pages.
    - **Delete**: After selecting events, delete action on bottom toolbar can be performed for all selected items.
- **Username**: The login name of the user.
- **Group list**: Group list in which is user assigned.

- **Privilege level**: The privilege level level determines whether a user is only allowed to control some basic aspects of the M&C system or whether the operator is allowed to change everything. Higher numbers mean higher privileges or more comprehensive control.
- **Functions**: Action icons for managing the user:
  - **Edit**: Pencil icon to modify/edit the user.
  - **Delete**: Trash can icon to delete the user.

Bottom toolbar

- **Delete selected users**: A button to delete selected users/rows from checkbox column.
- **Create user**: + button is used to create new user.
- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.20.1 Functionality

Create user

- Click the + button at the bottom of the table to add a new user.
- Input the required details.

Edit user

- Click the *pencil icon* in the Functions column to modify user details.
- Adjust the settings and save.

Delete user

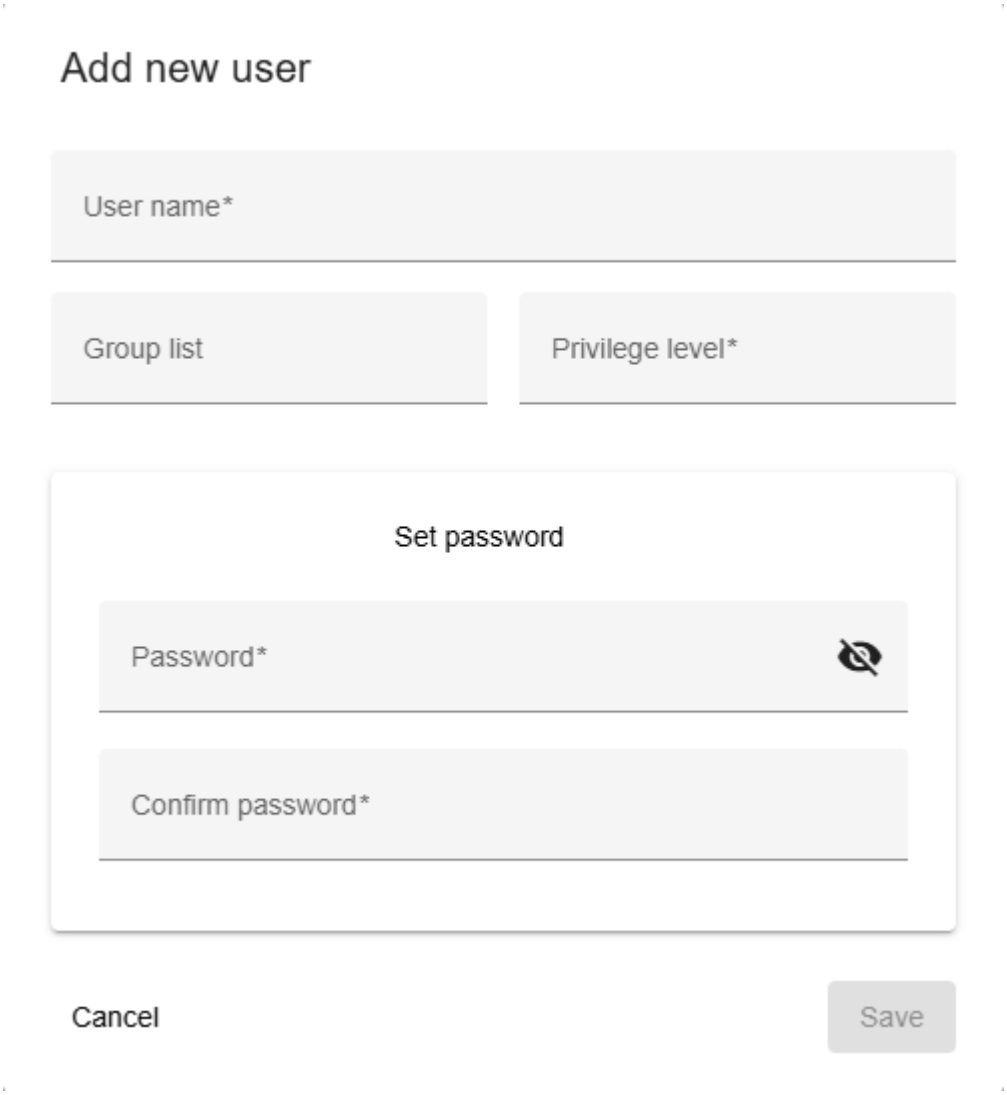
- Click the *trash icon* in the Functions column to delete an user.
- Confirm the action in the pop-up dialog to delete the user.

Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.20.2 Dialogs

Add/edit user dialog



- **Trigger**:
  - when you click on the "+" button on the bottom toolbar to create new user or
  - an on *pencil icon* to edit in the menu column
  - The dialog title clearly indicates whether you are creating a new user, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Username**: The login name of the user with which the user log in.
  - **Group list**: User group list.
  - **Privilege level**: Set privilege level for controlling user access. Must be greater then 1.
  - **Set password/ Change password**: If you want to edit or reset an existing user password, change the password displayed instead of *Set password* and simply expand it to change password.
    - **Password**: Set use password.
      - **Show password**: Click on visibility/eye icon to show or hide (default) password.
    - **Confirm password**: Confirm password.

- **Actions**:
  - **Save**:
    - This saves modifications.
    - If an error occurs in the input fields, the button is deactivated.
  - **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column or trash icon from bottom toolbar.
- **Content**: Asks the user to confirm whether they want to delete the selected user(s).
- **Actions**:
  - **Yes**: Deletes the user(s) permanently.
  - **Cancel**: Cancels the deletion action.

2.21 User settings

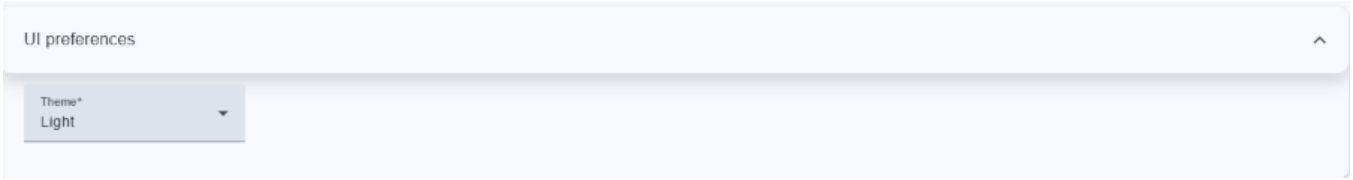
The User Settings page allows users to customize their experience and configure preferences. All changes made on this page are saved directly in the browser's storage or in the Database, ensuring personalized settings persist across sessions.

Title: Title of page.

Save mode: You can save your settings in the browser cache (default) or in the database.*The database or default settings specified by the admin (unless the setting is locked) are overwritten if the user has saved settings in the browser cache.* **Global settings**: Navigate to Global settings (admins only) **Key features**: Clicking on title will expand that section

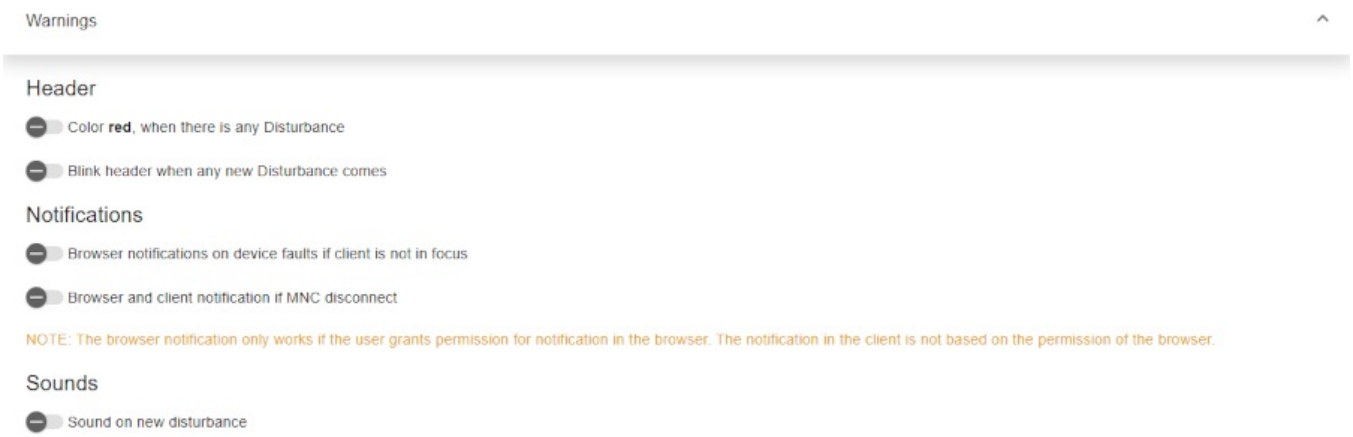
- UI preferences
- Warnings
- Edit take mode
- User screen
- Navigation
- Menu shortcuts
- Table settings

2.21.1 UI preferences



Theme: It an drop-down to choose between light, dark or automatic. Automatic option will adapt your system theme.

2.21.2 Warnings



Top

- Section title
- Icon to expand or collapse section

Header

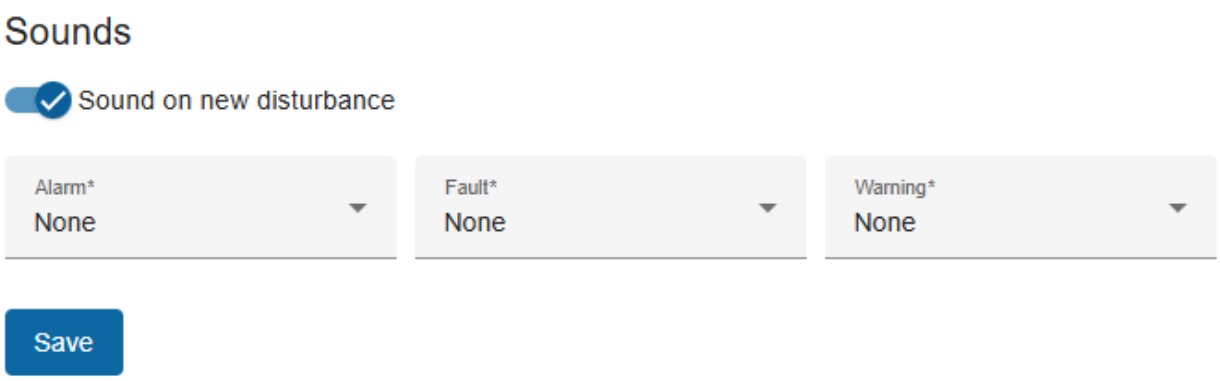
- Color red, when there is any Disturbance
- Blink header when any new Disturbance comes

Notifications

- Browser notifications on device faults if client is not in focus
  - If user client is not in focus or user find hisself in other tab, a notification will informs you. *NOTE: will not work on tablet or phone*
- Browser and client notification if MNC disconnect
  - If an MNC got disconnected you will be informed either with web browser notification or with snackbar.

Sounds

- Sound on new disturbance

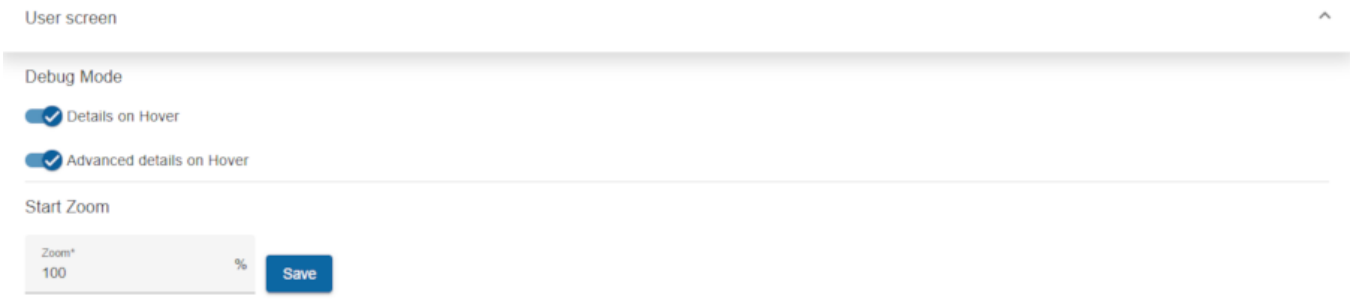


- If checked you can set different sounds for different warnings.
- **Save**: To save settings.

2.21.3 Edit take mode

Configure the edit take mode for device windows. If turns on, device window parameters are disabled. You can click on edit button to start editing parameters on device window and clicking on take button will save the edited values.

2.21.4 User screen



Debug mode

Enables additional debugging features, such as hovering over elements to display detailed information for debugging purposes.

- **Details on hover**: display some information in the bottom left corner when the mouse pointer is moved (hovered) over a display element (e.g. display element type, messageId, value and device mode) on the user screen-
- **Advanced details on hover**: same as *Details on hover*, but this displays more information.

Start zoom

Users can define a custom zoom level (in %) for the user screens. Once set, the user screen

will always display at the specified zoom level.

- **Zoom**: Type wished value. Default is 100%.

Save: Will save the zoom level.

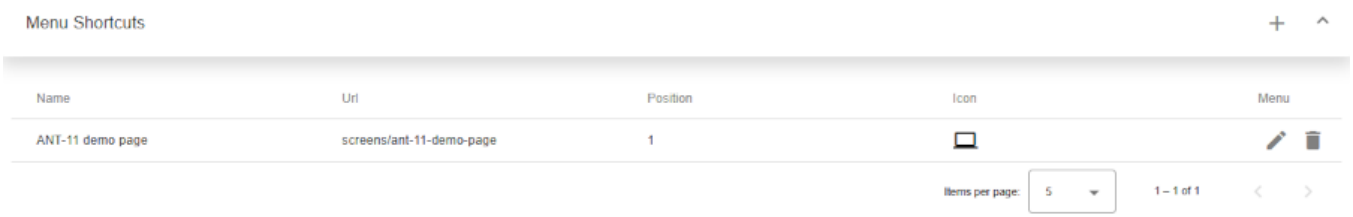
2.21.5 Navigation

Provides the option to set a specific page as the home page for easier navigation after user successfully logged in.

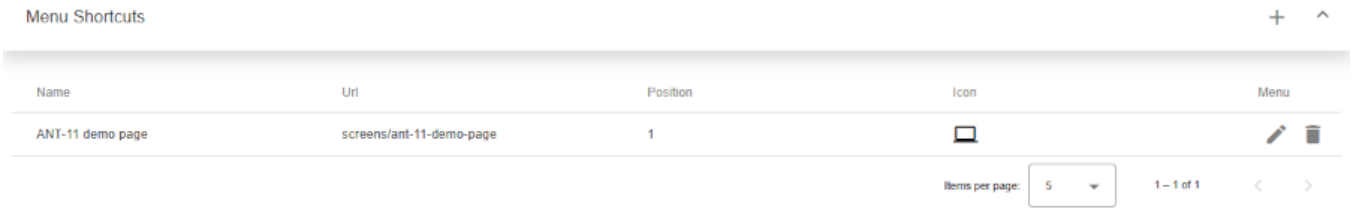
Set default url: Just type your whished url.

Save: Will save the url

2.21.6 Menu shortcuts



Allows users to add custom links under *Shortcuts* in the sidebar with other menus for quick access (in the image above) to frequently used pages or resources.



Top

- Left-side title of table.
- Plus (+) button to create new shortcut.
- Icon button to expand or collapse table

Table columns

- **Name**: The name of the created shortcut.
- **Url**: Saved url of shortcut.

- **Position**: Position of the shortcut.
- **Icon**: Shortcut icon.
- **Functions**: Action icons for managing the shortcut:
  - **Edit**: Pencil icon to modify/edit.
  - **Delete**: Trash can icon to delete.

Bottom toolbar

- **Pagination controls**: Allows users to set the number of items displayed per page and navigate between pages.

2.21.6.1 Functionality

Create shortcut

- Click the + button at the top of the table to add a new shortcut.
- Input the required details.

Edit shortcut

- Click the *pencil icon* in the Functions column to modify shortcut details.
- Adjust the settings and save.

Delete shortcut

- Click the *trash icon* in the Functions column to delete an shortcut.
- Confirm the action in the pop-up dialog to delete the shortcut.

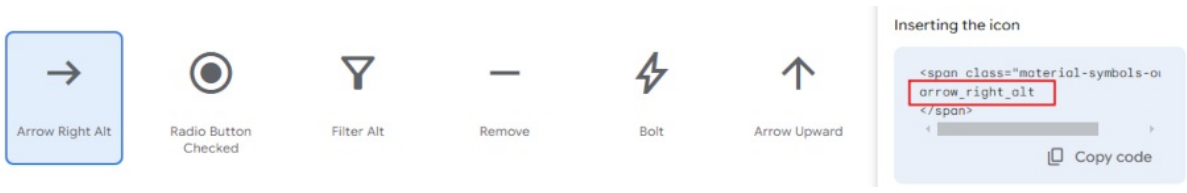
Pagination

- Adjust the number of rows displayed per page using the dropdown menu (e.g., 5, 10, or 25).
- Use the navigation arrows to move between pages of the table.

2.21.6.2 Dialogs

Add/edit shortcut dialog

- **Trigger**:
  - when you click on the "+" button on the top toolbar to create new or
  - an on *pencil icon* to edit shortcut in the menu column
  - The dialog title clearly indicates whether you are creating a new, editing an existing one, ensuring clarity in the action being performed.
- **Fields**:
  - **Menu name**: The name of the menu a user want to have.
  - **Url**: A url on which user want to have quick access.
  - **Icon**: The icon name can be chosen from *Google icons* and should be entered (lower case).



    - As soon as you have selected a icon, a window with the details appears on the right-hand side. Select or write the name of the icon in the *Inserting the icon* area, as in the example above with the red frame.
    - If the icon is not displayed after saving the shortcut, this means that the icon is not available.
  - **Position**: Enter the position of shortcut to have the shortcut in right order.
- **Actions**:
  - **Save**:
    - this saves modifications.
    - If an error occurs in the input fields, the button is deactivated.

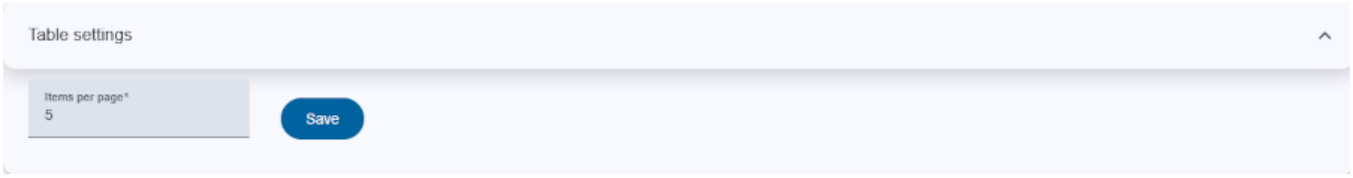
- **Cancel**: Closes the dialog without saving.

On delete confirmation dialog

- **Trigger**: Opens when you click the trash icon for an key from menu column.
- **Content**: Asks the user to confirm whether they want to delete the selected shortcut.
- **Actions**:
  - **Yes**: Deletes the shortcut permanently.
  - **Cancel**: Cancels the deletion action.

2.21.7 Table settings

Provides the option to set settings on tables of the app.



Items per page: Set number items that table should show.

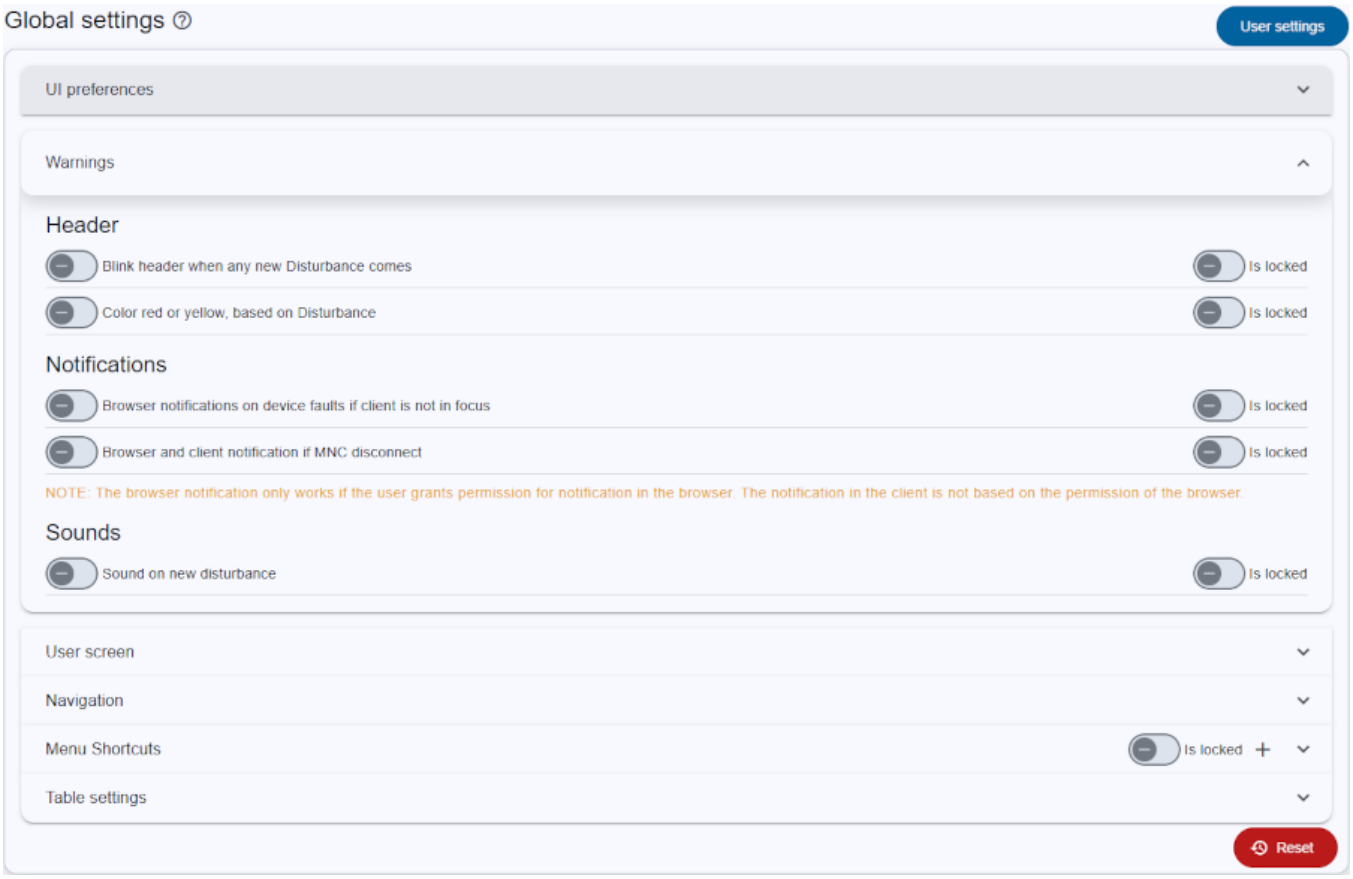
Save: Will save the setting.

2.21.8 Reset

2.22 Global settings

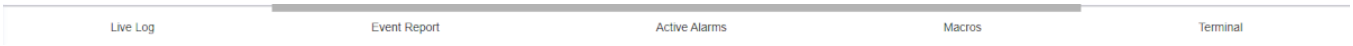
The Global Settings Page allows administrators to define and enforce default configurations across all users in the application. It shares the same structure and layout as the User settings, providing a familiar interface with the added ability to lock specific settings. Only users with administrative privileges can access this page.

Settings configured here act as defaults for users who haven’t explicitly saved their own preferences. Each setting includes a lock toggle. When a setting is locked, users cannot modify it in their personal settings, and the locked value is automatically applied by the application.



2.23 Bottom panel

The bottom section of the webpage, referred to as the Bottom Toolbar, offers quick access to essential features and logs for system monitoring and interaction. The Bottom Toolbar is available throughout the application, ensuring users can quickly interact with features at all times. The bottom toolbar is expandable with selected tab.



It contains the following tabs:

- **Live log**: displays real-time logs and M&C activity.
- **Event report**: Shows a detailed report of recent events in the system.
- **Active Alarms**: Lists all currently active alarms or notifications.
- **Macros**: Provides access to automated scripts or macros.
- **Terminal**: Offers a direct terminal interface for advanced system commands.

2.23.1 Interaction features

Persistent accessibility

- The bottom toolbar is always present and can be accessed from any page in the app.
- Simply hover on the bottom toolbar to reveal the toolbar's resize handle.

Resize handle

- **Drag to resize**: Once the handle is visible, click and drag it upwards to expand the toolbar or downwards to minimize it.
- **Double-Click**: Double-clicking the resize handle will instantly expand the toolbar to full screen and by full screen double-clicking the resize handle minimize the toolbar instantly.

2.23.2 Live log

The Live Event Log window shows the recent event messages as they are received from one of the M&Cs. The Live Event Log window shows the messages in the order of their arrival at the M&C. The window can be resized, scrolled and always shows the last 25 messages.

The sort order of the event list (newest first / oldest first) may be toggled using the*Sort by* dropdown on the right hand side. The default sort order is determined from a parameter set in the "client.properties" file located at the M&C server. Setting the key "gui.eventSortOrder" to "true" or "false" defines the default behavior of the live event log window.

Table columns

- **ID**: The event ID. This is a unique number the software assigns to each event which arrives at the M&C.
- **TIME**: The time stamp when the event was originated
- **NAK**: The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - : This is an informational message (lowest priority)
  - : This is a fault message (medium priority)
  - : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **SOURCE**: The primary message source. This usually is the ID of the MNC which issued the message. Messages marked with 'USER' or 'MNC' in this column.
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.

- **MESSAGE**: The message text.

Right side toolbar

- **Sort table**: The table content can sort by descending and ascending.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.
- **Acknowledge fault**: This button may appear if you want to acknowledge messages marked with the red exclamation mark. Simply select the row with red exclamation mark with mouse click. The exclamation mark will then disappear and a confirmation message will be added at the end of the log.

2.23.3 Event report

The Event Report window lets you generate filtered reports from the event database maintained by the NMS. This database contains all event (fault) messages issued by the MNCs during the last months (the exact depth of event storage is configurable during system installation).

The event report shown in the table gets automatically updated when you change the time range or apply other filter to display or the sort order. This lets you combine various filter specifications without being interrupted by the generation of intermediate reports.

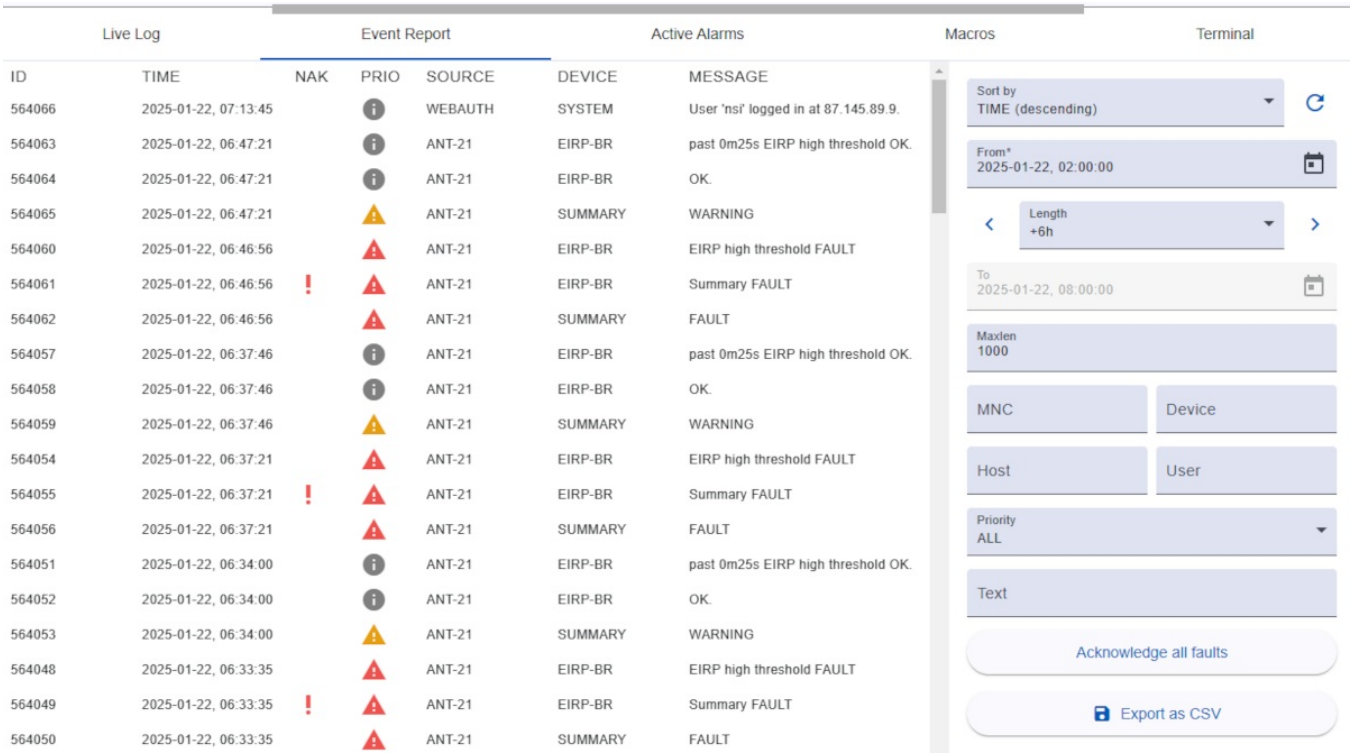


Table columns

- **ID**: The event ID. This is a unique number the software assigns to each event which arrives at the M&C.
- **TIME**: The time stamp when the event was originated

- **NAK**: The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - : This is an informational message (lowest priority)
  - : This is a fault message (medium priority)
  - : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **SOURCE**: The primary message source. This usually is the ID of the MNC which issued the message. Messages marked with 'USER' or 'MNC' in this column.
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.
- **MESSAGE**: The message text.

Filter functions

Located on the right side of event report tab. All filter can be combined with each other.

- **Sort by**: The table content can sort by descending and ascending of it's time.
- **Reload button**: Reload the table content to get latest data.
- **From**: Define the start time. If you click on the calendar icon on the right-hand side, a calendar view for selecting the date is displayed. The time must be entered manually.
- **Length**: The length is a drop-down menu that allows you to select duration. If you click on the arrow button on the right or left, the hours are added (on right arrow) to or subtracted (on left arrow) from the start time.
- **To**: Define the end time. Normally this input field is disabled to activate, please select 'select end time' from length dropdown.
- **Maxlen**: Define the maximum number of events that a report may contain. There is a default value for this (1000 in most cases), which represents a good compromise between performance and the number of events displayed simultaneously. You can change this limit according to your needs, but remember: reports with millions of events can take some time.
- **MNC**: This filter lets you filter with the M&C name.
- **Device**: This filter lets you filter messages along their device (secondary source) which caused the message.
- **Host**: The host name filter lets you restrict the report to messages originated from a certain client PC. Login messages and messages logging the active change of a parameter are stored with the origin (hostname or IP4 address) of the event.
- **User**: Lets you restrict the report to messages originated by a certain user. Login messages and messages logging the active change of a parameter are stored with the name of the user who is responsible for this event.
- **Priority**: The priority selector defines the priority (or type) that an event must have in order to be listed in the report. The selection *ALL* shows all events. *FAULT* contains events that have at least fault priority. *ALARM* only shows alarm messages (see the description

of alarm priorities above). The NOT ACK selection shows all messages that (still) need to be acknowledged.
- **Text**: Filter with the message text.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.
- **Acknowledge fault**: This button may appear if you want to acknowledge messages marked with the red exclamation mark. Simply select the row with red exclamation mark with mouse click. The exclamation mark will then disappear and a confirmation message will be added at the end of the log.
- **Export as CSV**: This button downloads the current event report, as displayed in the table, as a CSV file.

2.23.4 Active alarms

The Active alarms tab lists all actually pending faults and warnings in the M&C. The Table automatically updates its content as the faults appear or disappear.

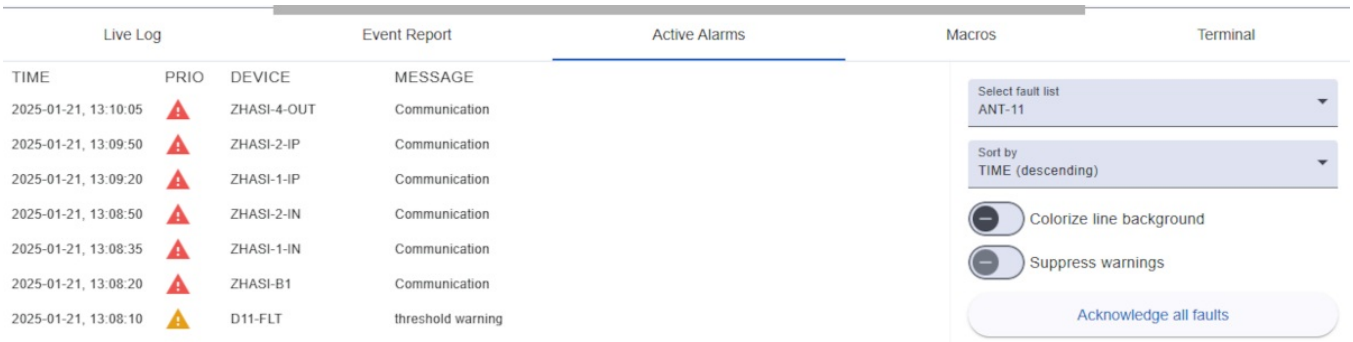


Table columns

- **TIME**: The time stamp when the event was originated
- **PRIO**: The message priority. The message priority decides on the urgency, the message gets delivered from the M&C to the client. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - : This is an informational message (lowest priority)
  - : This is a fault message (medium priority)
  - : This is an alarm message (highest priority, the MNC tries to deliver this immediately, regardless of the dial timing settings)
- **DEVICE**: The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a MNC which caused the event.
- **MESSAGE**: The message text.

Right toolbar

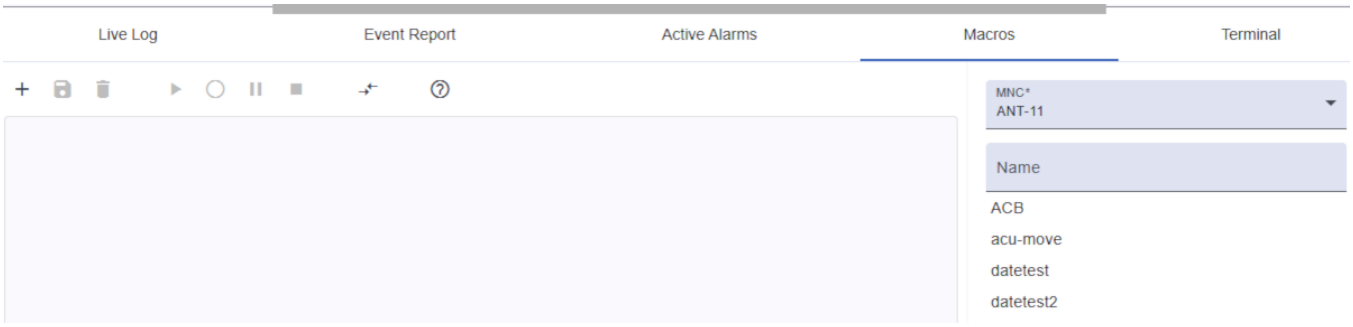
The toolbar buttons of the window let you select how the fault list displays the active faults:

- **Select fault list**: Select an MNC name from the drop-down list for which you want to check faults. Primary M&C is selected as default.

- **Sort by**: The table content can sort by descending and ascending of it's time.
- **Colorized the background**: If checked, the background of each line gets background color as the icon red for faults, yellow for warnings.
- **Suppress warnings**: If checked, the table content suppresses all warnings and show only faults only. Otherwise faults and warnings are displayed.
- **Acknowledge all faults**: Acknowledge all faults marked with the red exclamation mark.

2.23.5 Macros

The sat-nms software includes a powerful macro recording and playback feature which lets you automate complex equipment settings to one mouse click. The following chapters describe the user interface which give access to this feature.



Top toolbar

You must select an M&C from the right side toolbar, so that this toolbar can be displayed.

- **Add macro**: On click on + button you can create the new macro.
- **Save macro**: If you made some modifications you can save by clicking on this button.
- **Delete macro**: Selected macro can be deleted.
- **Play macro**: Play the selected macro.
- **Record macro**: You can start recording the macro. On toolbar of app the record icon will be displayed.
- **Pause/resume recording**: You can pause or resume recording.
- **Stop recording**: Stop the macro recording.
- **Compare macros**: To compare macros, please select macros from the macro list on the right-hand side. Both macros are displayed next to each other. The “Compare” button is displayed at the bottom of the window. It triggers a dialog that displays a better comparison. Clicking again on button will disable compare mode.

Right side toolbar

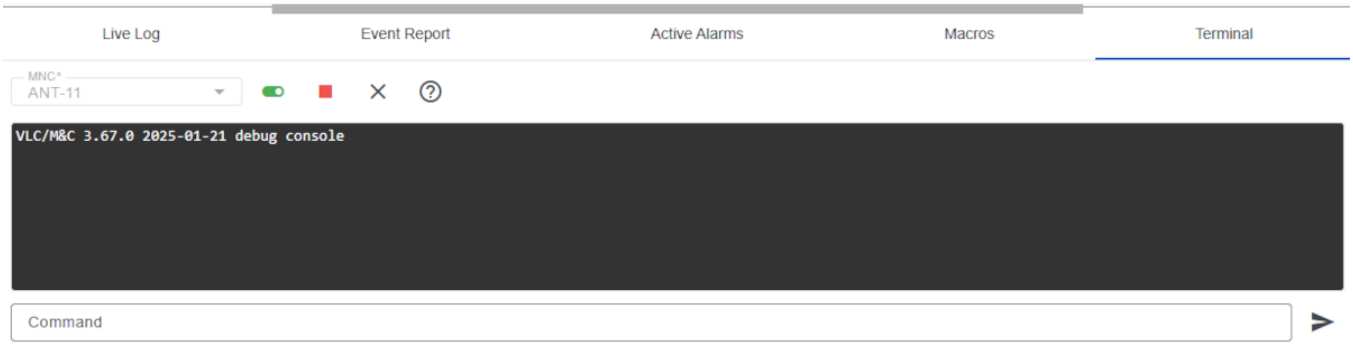
- **MNC**: Select an M&C to load all stored macros or perform other actions.
- **Name**: On creating new macro name should be entered, otherwise it works as a filter also. You can type the name of the macro to find the saved macro.
- **Macro list**: Once you selected an M&C, a macro list will be displayed under the name field. Simply select macro to perform actions described in above toolbar section.

Editing area: The macro editing area is located under the top toolbar. You can also edit the

saved macro by simply typing as you would in a text editor. This area is also used to compare macros.

2.23.6 Terminal

The terminal tab gives you access to the debug console of a M&C system. This is a valuable tool to inspect the low level communication to devices which are controlled by the M&C.



The terminal window consists of a display area which shows the debug output of the M&C and a command entry field at the bottom of the window. The tool bar at the top gives you access to the following functions:

1. **MNC**: Select an M&C to which you want to start the terminal connection.
2. **Switch**: After selecting an M&C you have to toggle this switch to start connect or disconnect the terminal connection. Green means terminal is connected.
3. **Stop/play icon**: It pauses or resumes the terminal session.
4. **Close icon**: It's clear the terminal display area.

After the terminal display area there are 'Command' input and 'send' icon. You can just type in command input and press 'enter' key to send command.

Debug commands

At the entry line at the bottom of the window you can enter a number of debugging commands which may help to test e.g. new device drivers. The following commands may be entered:

- get <msg-id> --- Get the current value of an parameter with the ID <msg-id>
- set <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would entered by the user
- loop <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would read from the device
- touch <msg-id> --- Force the parameter with <msg-id> to be read from the device with the next driver cycle
- threads --- Show a comprehensive list of device threads
- thread <n> --- Show details of thread number <n>
- peers --- Show peers / list of connected clients
- peer <n> (view|verbose|silent|kill) --- Details and functions of peer number <n>
- restart --- Restart MNC service (same as restart button in client)
- shutdown --- Stops MNC service. Attention: service has to be restarted on the command

line

- limit / nolimit --- make the 'debug.log' file unlimited or limited in size. The MNC always starts with the 'debug.log' file limited in size.
- csm-debug --- toggles the debug output of**all** CSM spectrum analyzers ON/OFF. The debug output appears in the `.panic.log` file. Remarks:
  - Be careful to switch csm-debug off again after using it. It remains on when the debug terminal window is closed unless it has been switched off before. This may cause a large amount data in the `.panic.log` file.
  - Using csm-debug only gives meaningful information if there is only one spectrum analyzer running in the system. If you have multiple spectrum analyzers configured, be sure to set all but the one you want to debug to OUT-OF-SERVICE.

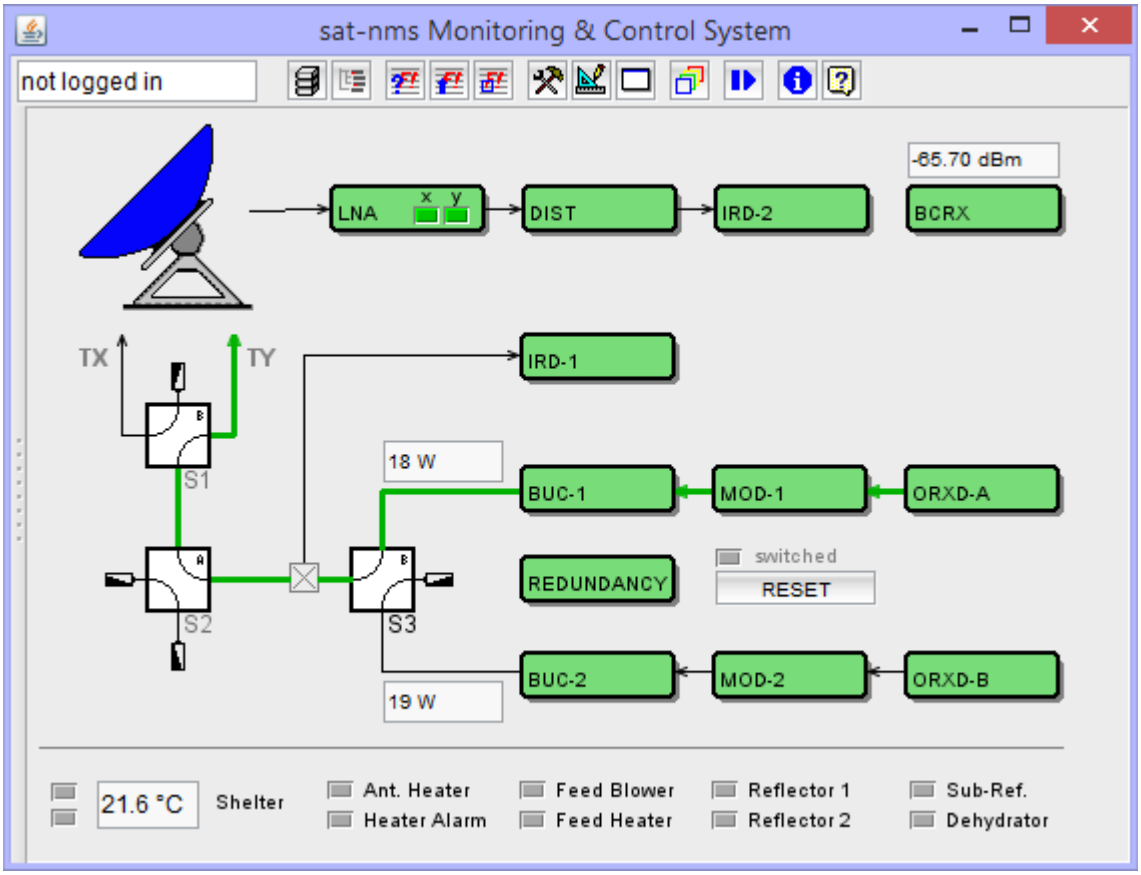
3 Legacy Java Client Operation

This section describes the functionality and the usage of the sat-nms M&C software when operated with the legacy Java client. The following chapters of the manual describe each function and each user interface screen in detail.

The satnms Java client is a legacy user interface to the software, it has been replaced by the newer Web based client. The documentation for the Java client is still provided for older M&C installations of if an installation explicitly uses the legacy client for some reason.

3.1 The M&C Main Window

The M&C main window forms the main entry point to the M&C user interface. As most of the subsidiary M&C windows, it is widely user configurable. In many cases the M&C main window will contain a block diagram showing the state of the devices operated by the M&C like shown in the example below.



In this block diagram each piece of equipment controlled by the M&C is represented by an icon in the window. The color (and/or shape) of the icons signals the state of each device. A double click to a icon opens the Device window for this device, a unified M&C user interface to it.

Tool-bar Functions

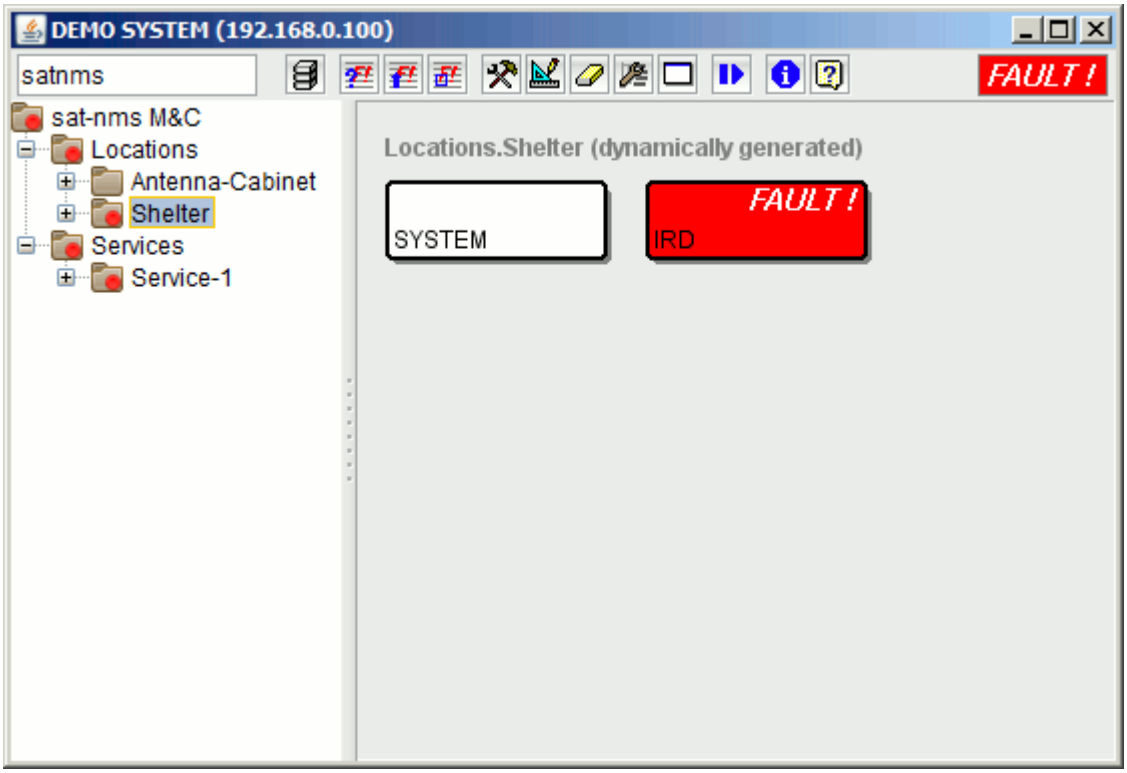
- --- The left field at the window's tool-bar shows the login-name of the user who is actually logged in. You may click with the mouse to this field, the program pops up the login/logout dialog.
- --- Shows a drop down list with all devices installed in the M&C system. Selecting a device from this list opens it's Device window .
- --- Opens the M&C Tree View Window .
- --- Opens the Event Report window .
- --- Opens the Live Event Log window .
- --- Opens a list of currently active faults.
- --- Opens the Device Setup window to configure the device setup of the M&C system.
- --- Opens the Screen Editor window to modify the layout of this M&C window.
- --- Opens a terminal window which connects to the debug / test port of the M&C server.
- --- Opens the User Management Window . You must be logged in with privilege level 150 or higher in order to open this window.
- --- Opens a file select dialog which loads or saves the window layout of the application.
- --- Opens the Macro Management window which controls recording and replay of equipment setting macros.

Remarks

- Touching a device icon with the right mouse button opens a context menu offering frequently used functions.
- As this window is widely user configurable, a particular M&C main window need not to look like the example shows here.
- Devices may be symbolized by other icons than shown here.

3.2 The M&C Tree View Window

The M&C Tree View Window is much like the M&C Main Window. It provides the same functions as the latter and adds a tree navigation bar at the left side of the window. M&C Tree View Window may used as a stand alone client window or launched from the M&C Main Window.



By default, the right side of the window shows the default user screen. The left part of the window shows the tree of subsystems and the devices attached to them. Each device node is marked with its fault state (no mark=OK, yellow mark=WARNING, red mark=FAULT). Subsystem nodes also show fault marks, summed up from the devices and subsystems contained in it.

Clicking on a node of the tree in the left part of the window changes the contents of the right part:

- If the node stands for a device, the device screen for this particular device is opened and embedded into the right part of the window.
- If the node stands for a subsystem, the right part of the window becomes a user screen, individually defined for this subsystem. This screen may be edited in the sat-nms GUI editor. If no such screen exists, the software dynamically creates screen containing all devices attached to this node in a grid.

The Tree View Editor may be used to edit the subsystem tree and the devices attached to each node.

Tool-bar Functions

- --- The left field at the window's tool-bar shows the login-name of the user who is actually logged in. You may click with the mouse to this field, the program pops up the login/logout dialog.
- --- Shows a drop down list with all devices installed in the M&C system. Selecting a device from this list opens it's Device window .

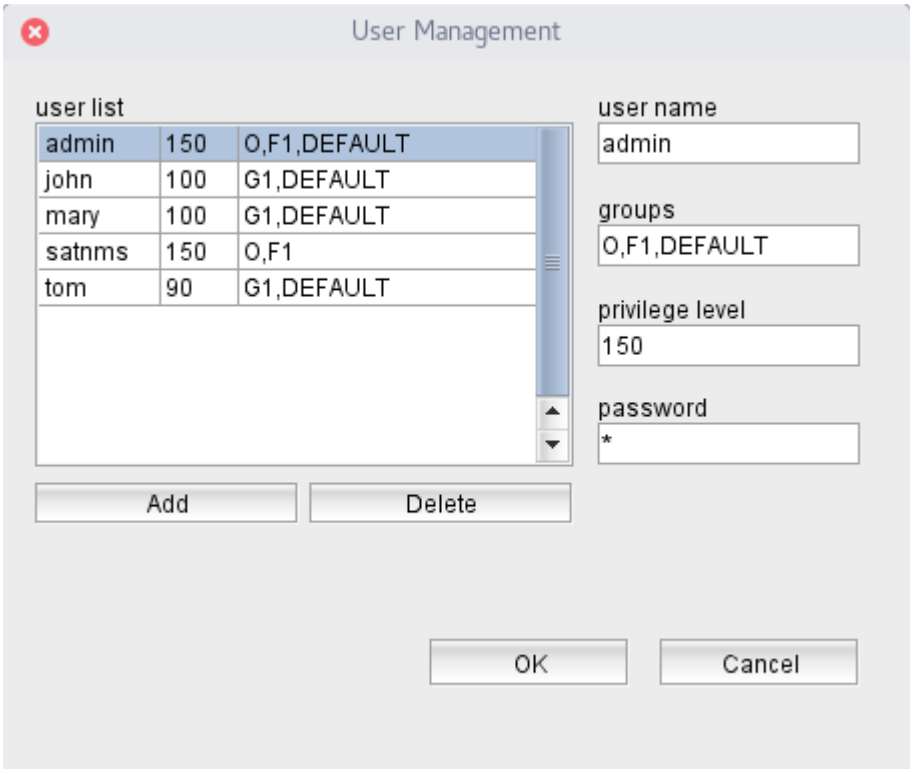
- --- Opens the Event Report window .
- --- Opens the Live Event Log window .
- --- Opens the Device Setup window to configure the device setup of the M&C system.
- --- Opens the Screen Editor window to modify the layout of the actually displayed user screen. If the screen is dynamically generated, it will be stored at the server's disc and will no longer be updated automatically.
- --- Deletes the user screen actually shown. This reverts the screen to the dynamically generated version provided by the software.
- --- Opens the Tree View Editor .
- --- Opens a terminal window which connects to the debug / test port of the M&C server.
- --- Opens the User Management Window . You must be logged in with privilege level 150 or higher in order to open this window.
- --- Opens the Macro Management window which controls recording and replay of equipment setting macros.

Remarks

Tree nodes may be expanded or collapsed either by clicking to the '+' marks in between the nodes or by double-clicking the node to expand/collapse.

3.3 The User Management Window

The user management window lets you manage the list of users which are allowed to operate the software. You need to be a "super user" with privilege level 150 or higher to open the user management window.



The user management window shows a list of all users known to the software. You may click to a user in the list to show the properties of this user in the entry fields right of the list.

- *user name:* The login name of the user. You cannot change the name of an existing user because this name identifies the user in the sat-nms software. You can however change the name in this field and press the button "Add" below the user list to create a new user definition.
- *groups:* This field takes a comma separated list of group identifiers. With the normal M&C software groups are meaningless, with the NMS software or the VioIP software however groups control which VLCs or which equipment a particular user may access. To change the group membership of the selected user simply edit/overwrite the group list and press enter.
- *privilege level:* The privilege level controls if a user may only control some basic aspects of the M&C system or if the operator may change everything. Higher numbers mean higher privileges or more comprehensive control. The privilege level numbers are often unique to the particular installation. Many installations of the sat-nms software however use the following standard levels:
  - 150 = super user, may control everything including all configuration tasks.
  - 100 = general user, may control all aspects of all devices, but may not change the sat-nms software configuration.
  - 90 = user with limited access to some devices or parameters only. Makes sense only together with a default screen wich gives no general access to all devices.To change the privilege level of a user, enter the new level value and then press enter.
- *password:* This field always displays a single asterisk, even as a super user you cannot see a user's password. You can however set a new password for the user simply by typing in into tge entry field and pressing enter.

To add a new user to the list, type the name of the user in the "user name" field and click the "Add" button. This takes the actually selected user as a template and creates a new user with the group memberships and privilege level of the template. The new user has no valid password set, you have to set an initial password for the new user explicitly by typing it into the "password" field. The user may change this password later.

To delete a user from the list, select the user to delete and click the "Delete" button below the user list.

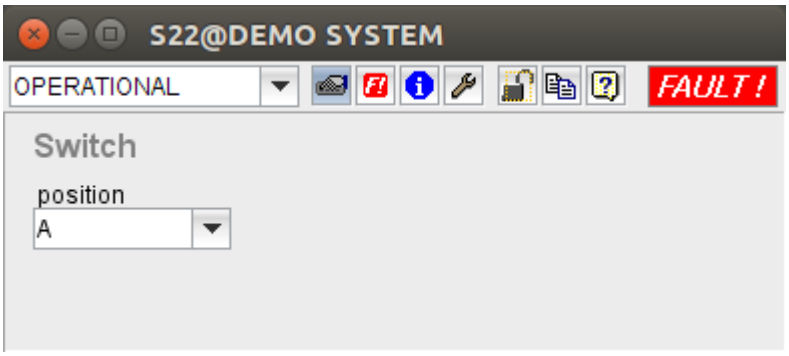
The software does not change the user database until you leave the user management window with the OK button. By pressing "Cancel" you can abort the edit process at any time.

Changed user permissions and credentials are valid with the next login of the user. Users currently remain with their privilege level and group assignments they had at the time they logged into the sat-nms software.

3.4 The Device Windows

The software offers a Device window for each type of equipment it supports. Device windows are launched by double clicking to the device icons in a M&C user interface window . It shows all parameters read from the device and permits to alter each writable parameter. For

clearance, the parameters are grouped to pages, you can switch between the pages using the tool-bar.



Writable parameters are shown with a light gray background, read only parameters appear with the standard gray background of the window. Choice parameters are changed immediately when you alter the selection. If you edit a text parameter, the field is shown with a yellow background. Once you press ENTER or you leave the field, the changed value gets transferred to the device. To cancel a parameter setting press ESC. The old value gets restored in the field and the background color returns to light gray. No parameter value is sent to the device in this case.

Tool-bar Functions

- --- The operation mode selector displays and sets the device operation mode. Device operation modes are:
  - **OPERATIONAL** --- This is the normal operation mode.
  - **FAULT-SUPPRESSED** --- The M&C server normally polls and controls the device, but does not generate a device summary fault if the is a fault condition with this unit.
  - **OUT-OF-SERVICE** --- The M&C server does not try to communicate with the device at all. If you alter parameters of a device in this mode, this has no effect.
- --- Opens a Load Device Preset dialog which lets you select a formerly stored device preset and apply the stored setting to the device. The chapter ' Managing Device Presets ' gives more information about device presets and how to handle them.
- --- Opens a Store Device Preset dialog which lets you store the actual device settings at the server for later retrieval.
- --- Switches the window to the 'faults' page. This page displays all fault flags the software knows about this device. Moreover, you can view and change the event priority for each particular fault flag:
  - Click to a fault flag and hold the mouse button down to see the event priority actually set.
  - Use the context menu (right mouse button) to change the event priority.

Valid selections for the event priority are:

- **OFF** --- This flag is completely suppressed.
- **INFO** --- This flag is signaled as an 'informational' event.
- **WARNING** --- This flag is signaled as a 'warning' event.
- **FAULT** --- This flag is signaled as a 'fault' event.
- **ALARM** --- This flag is signaled as an 'alarm' event.
- --- Switches the window to the 'info' page. This page gives some information about the type of equipment controlled.
- --- Switches the window to the 'maintenance' page. This page lets you set some configuration parameters and gives access to the 'low level interface' of the device.
- --- Locks the operation of the device that no other user can change device settings. See paragraph device locking below for details. If locked, the button shows a closed red lock:
- --- Duplicates / clones this window. This is useful if you want to have more than one instance of a device window open at a time.
- --- Launches the web browser with the help page for this individual device type. For a list of device types supported by the software see the 'Device Driver Reference' .

Device locking

Clicking the button once locks the operation of the device. The button icon changes to . If the device is locked, no other user may change devices settings. Locking is done along the user name, hence if you are logged in a second time an another place, you can operate the device from there as well.

A second click to the lock button releases the lock. Only the operator who set the lock may release it. This need not to be done at the same place where the lock was set bot you must be logged in with the same user name.

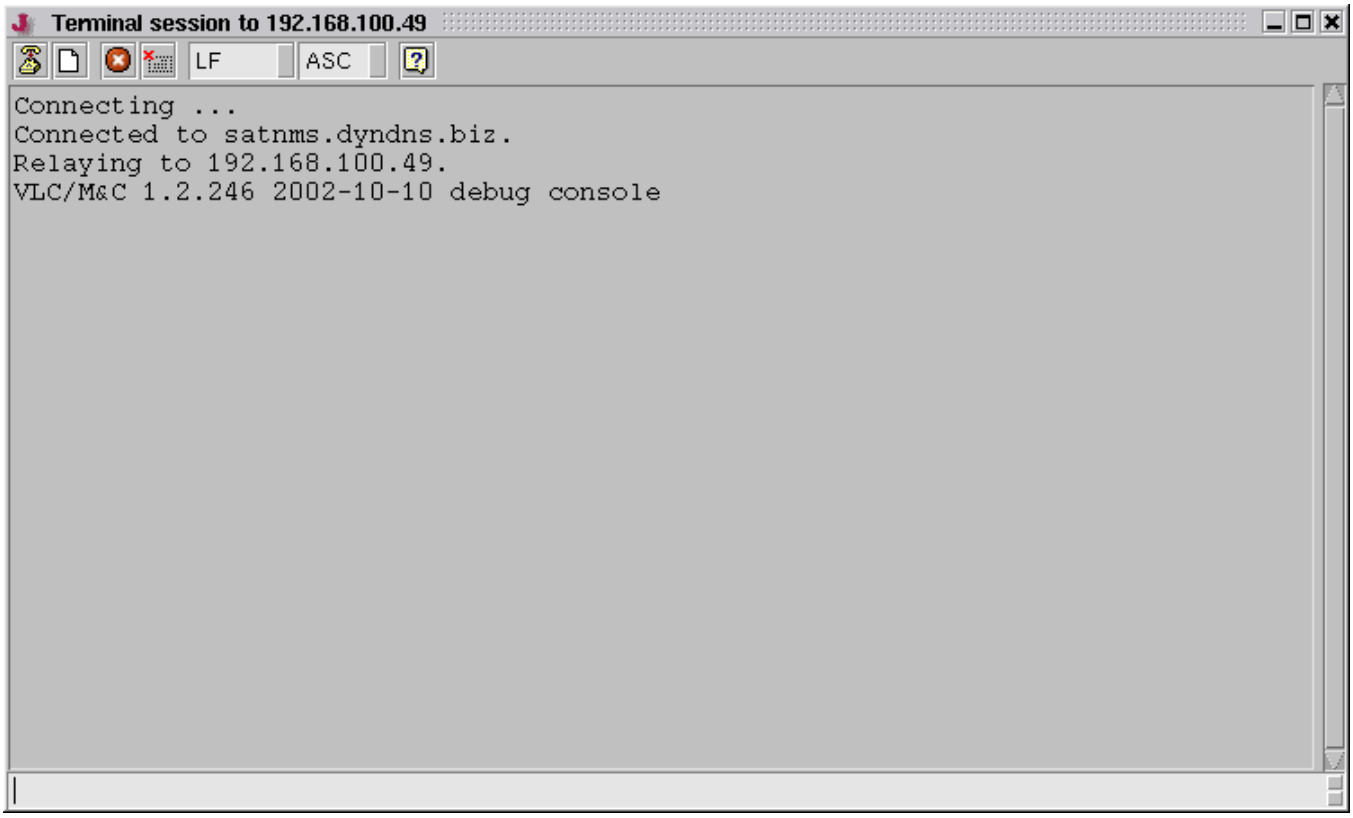
Administrators (with a privilege level of 150 or above) may release the lock in any case, also a restart of the sat-nms M&C server releases all pending device locks.

Remarks

- Units and unit factors displayed with a parameter are for information only. They are ignored when you enter a new value. Hence you cannot enter a value expecting 'MHz' in GHz by changing the unit multiplier.

3.5 The Terminal Window

The terminal window gives you access to the debug console of a VLC or M&C system. This is a valuable tool to inspect the low level communication to devices which are controlled by the M&C.



The terminal window consists of a display area which shows the debug output of the VLC and a command entry field at the bottom of the window. The tool bar at the top gives you access to the following functions:

- --- Disconnects from the VLC / M&C and closes the window (same as the 'q' command).
- --- Clears the window.
- --- Temporarily inhibits the output of data sent by the VLC / M&C.
- --- closes a 'term' connection to a certain serial interface.
- LF --- Selects the 'end of line' terminator appended to each commands entered to the command field at the bottom of the window. Possible selections are LF, CR, CR/LF, and ---.te .ts ASC Selects ASC or HEX mode. ASC is the normal terminal mode displaying printable characters as they are. HEX is the HEX-dump mode displaying each character as a 2-digit hexadecimal number. The mode affects the command entry, too.

Debug commands

At the entry line at the bottom of the window you can enter a number of debugging commands which may help to test e.g. new device drivers. The following commands may be entered:

- get <msg-id> --- Get the current value of an parameter with the ID <msg-id>
- set <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would entered

by the user
- loop <msg-id> <v> --- Set parameter with the ID <msg-id> to a value as it would read from the device
- touch <msg-id> --- Force the parameter with <msg-id> to be read from the device with the next driver cycle
- threads --- Show a comprehensive list of device threads
- thread <n> --- Show details of thread number <n>
- peers --- Show peers / list of connected clients
- peer <n> (view|verbose|silent|kill) --- Details and functions of peer number <n>
- restart --- Restart MNC service (same as restart button in client)
- shutdown --- Stops MNC service. Attention: service has to be restarted on the command line
- limit / nolimit --- make the 'debug.log' file unlimited or limited in size. The MNC always starts with the 'debug.log' file limited in size.
- csm-debug --- toggles the debug output of**all** CSM spectrum analyzers ON/OFF. The debug output appears in the `.panic.log` file. Remarks:
  - Be careful to switch csm-debug off again after using it. It remains on when the debug terminal window is closed unless it has been switched off before. This may cause a large amount data in the `.panic.log` file.
  - Using csm-debug only gives meaningful information if there is only one spectrum analyzer running in the system. If you have multiple spectrum analyzers configured, be sure to set all but the one you want to debug to OUT-OF-SERVICE.

3.6 The History File Viewer

The sat-nms M&C software provides a facility (a logical device) calledFile-Recorder which is capable to record arbitrary data like levels, antenna pointing etc. into a disc file. The *History File Viewer* is the user interface to inspect such files. The window can be accessed either from the File-Recorder's device window or can be used as a stand alone application as well (see below for details).

The window shows a diagram of the recorded data, the fields and buttons in the tool bar permit to modify the appearance of the diagram as needed. Clicking into the diagram shows a marker which displays a readout of the value at the selected screen position.

Top Toolbar

- --- Loads a file from a local disc. Opens a standard file select dialog. You may use this to display files you have archived on you local PC.
- --- Gets a file directly from a remote sat-nms M&C. When you started the*History File Viewer* from within the sat-nms GUI, it usually will already display the data you want to see on startup. If you started the file viewer as a stand alone application, you use this function to connect to a M&C system and get the data from there.
- --- Saves / archives the recorded data as a disc file to the local computer. A file select dialog is shown where you can select or enter the name of the file to save. The file format is text based, saved files can be re-opened with the open icon in the tool bar. The textual format of the file also permits to process the saved data with a spreadsheet program or with gnuplot.
- --- Moves the plot two divisions to the left, lets you see older data.
- --- Moves the plot two divisions to the right, lets you see younger data.
- --- Shows/edits the start time. The displayed time corresponds to the leftmost grid line in the diagram. You set the start time by changing the value with the keyboard and then pressing ENTER or you may double-click to the field which displays a small calendar window which lets you select a another date in a comfortable way. When you change the start time, you should consider, that the software always aligns the value that in fits into the grid of the time scale. This means if e.g. the time scale is 1 hour/div, a start time

entered gets always aligned to a full hour.

- --- Zooms in the time scale, shows a smaller section of the data more detailed.
- --- Zooms out the time scale, shows a larger section of the data.
- --- Zooms out to show the entire plot. Please note, that the largest available scale shows 720 days (30d/div). If the data you are displaying covers a larger time span, you will not see the complete data even after zooming out fully.
- --- toggles live view / normal mode (see paragraph "Live view mode" below for details).
- --- Opens the trace properties dialog window (see below for details).
- --- Prints the diagram as actually shown.
- --- Shows this manual page.

Bottom Toolbars

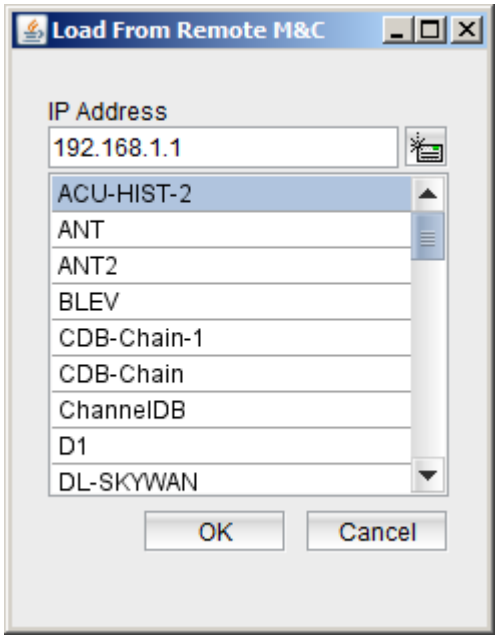
Below the diagram, the History File Viewer provides four toolbars to modify the scaling of the up to four traces shown in the diagram.

- --- The first control in each toolbar selects if the trace shall be drawn.
- --- Shows the color if this trace, clicking to the color fields opens a dialog which lets you change the color of the trace.
- --- Shows the name / description of the trace. When called directly from the File-Recorder device screen, the History File Viewer shows the IDs of the recorded parameters here, "Trace n" if the trace name is not known. Trace names may be set by the operator and saved on the M&C server, see the paragraph about adapting the diagram properties below.
- --- Moves the trace upwards one division.
- --- Moves the trace down one division
- --- Shows the y-scale reference level and the actual scaling. The y-reference level is the value corresponding to middle grid line.
- --- Zooms in the y-scale for this trace.
- --- Zooms out the y-scale for this trace.
- --- Zooms to show the min/max values (autoscale). This refers to the entire data, not only to the part actually shown.

Getting data directly from a M&C system

The history file viewer is capable to get the data to tisplay directly from a remote sat-nms M&C system. This feature can be used also if the viewer has been started as a stand alone application.

Clicking to the button in the tool bar opens the remote load dialog window. In this window, enter the IP address of the M&C computer you want to read from and click to the button right beside the entry field.

- 

--- The selection list field now shows 'connecting ...' and after a couple of the seconds, the files which look like File-Recorder data files are displayed in the the selection box. Select the file (the name of the File-Recorder logical device) you want to load an click to 'OK'.The histoy file viewer in this case loads *both* files belonging to the File-Recorder device if they are available. If you load the data from a File-Recorder named FREC1, the file viewer tries to get the files FREC.old.txt and FREC.txt and concatenates them in memory.

If the file recorder is not able to read from the remote M&C system, the 'connecting ...' notice remains visible permanently. In this case verify the IP address you entered, close the window and try again.

Live view mode

Beside for viewing stored log files, the *History File Viewer* can be used to display the logged data as it is acquired with an automated display update. By pressing the "live view" toggle button in the toolbar, the viewer changes to live view mode. In this mode the viewer shows up to 400 recent data points, automatically updated with an interval that is configurable separately from the standard interval at the File-Recorder device setup page.

Due to the automatic display update, the viewer and its autoscale functions behave slightly different in live view mode:

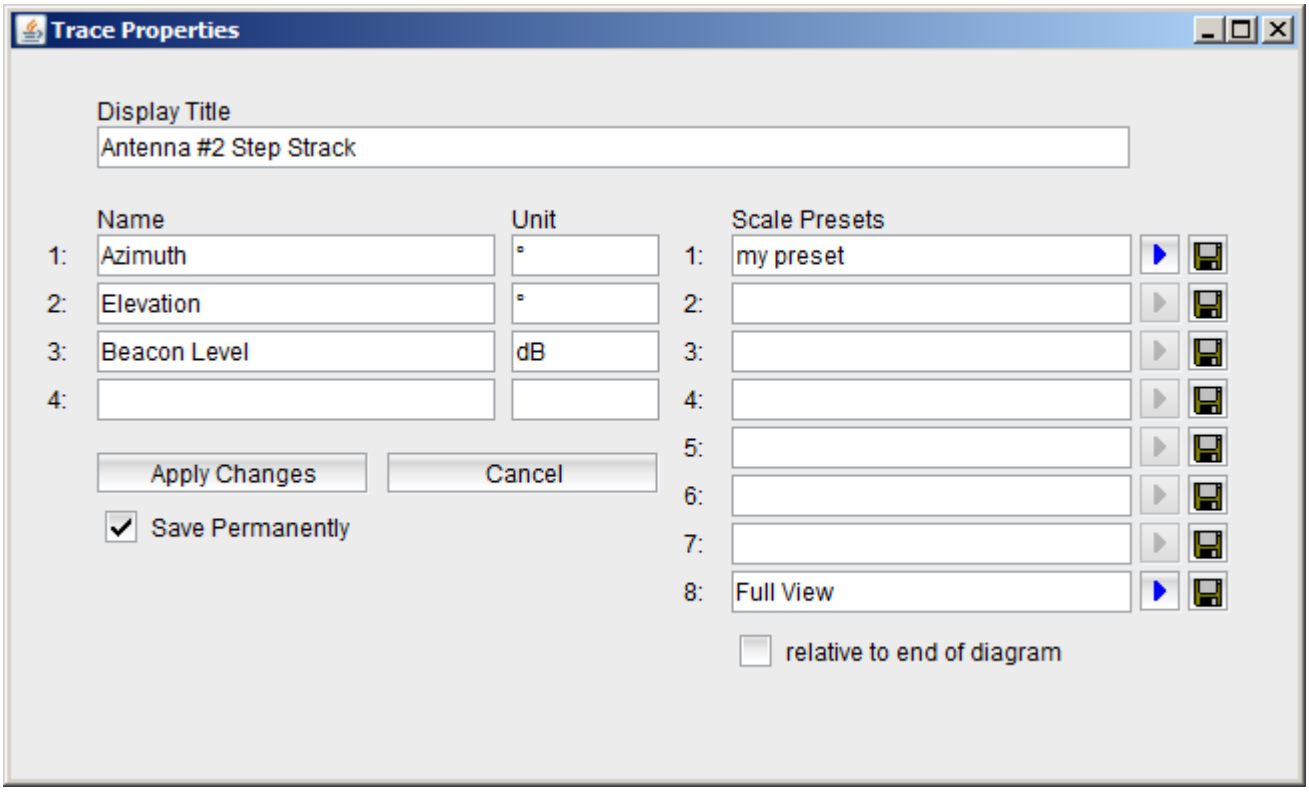
- The diagram gets aligned with the right edge of the frame with time axis autoscale and every time the display gets updated with new data points.
- When switching to live view mode, the time scale is adjusted once to show the entire (up to 400) live view data points.
- You may change the time axis scaling to your need with the zoom in/out buttons in the upper toolbar, the viewer won't overwrite this when the display gets updated.
- You also may change the time axis start value with the blue arrow buttons in the upper toolbar, but the viewer will revert this when the display gets updated.
- You cannot zoom or scroll outside the limited range of data points used for live view

mode. To view older data, switch back to normal mode.
- Loading or saving disc files is not possible in live view mode. If you try, a message is displayed and the viewer remains in live view mode.
- On the other hand, live view mode is not possible if the viewer shows the contents of a local file instead of the data from a File-Recorder device on a M&C.

To leave live view mode, click the "live view" toolbar button again. The viewer re-reads the full history data and performs an autoscale.

Adapting the diagram properties

The in the tool bar opens the trace properties dialog window. This dialog lets you set the name of each trace, the unit which shall be displayed with the scaling information fo each trace and a diagram title. The latter is shown in the diagram window title bar and also above the plot in the printout.



If you select the 'save' option, the trace properties are written to file named like the data file, but with the suffix '.frv'. The file is written to the same place where the data file was loaded from. When you load this data file the next time, the saved trace properties are used automatically.

The Trace Properties dialog also lets you save and recall up to 8 scaling presets. These presets may help you recall certain settings of y/t scaling.

- --- Stores the name of the preset, the color of each trace, if the trace shall be drawn or not and the actual scale settings to the given preset memory. The presets are stored together with the trace names in the '.frv' file on the M&C server.

- --- Recalls this preset, restores the trace color, the trace enable state and the scale settings from the stored values. If the preset wasn't set before, the preset recall button is inactive.

If you check the mark below the preset list before recalling a preset, the start time of the diagram is set relative to diagram end. This means, the start time stored in the preset is ignored, the start time is set to show the recent data in the diagram.

Zooming to the cursor position

When you zoom the time axis of the diagram using the ool bar buttons, the viewer tries to keep the start time of the diagram constant. If you want to have a more detailled look on a certain event, the event you want to zoom to will probably move out tof the visible part of the display. To zoom to certain position on the time axis, hold down the SHIFT key and click to the position you want to zoom to. Holding the CTRL key while clicking works the other way round, the plot gets zoomed out.

Starting the viewer as a stand alone application

Assuming your sat-nms client software is installed at 'c:\satnms', the command line to start the history viewer is for a windows based computer:

java -cp client.jar satnms3.gui.FiledRecordFrame

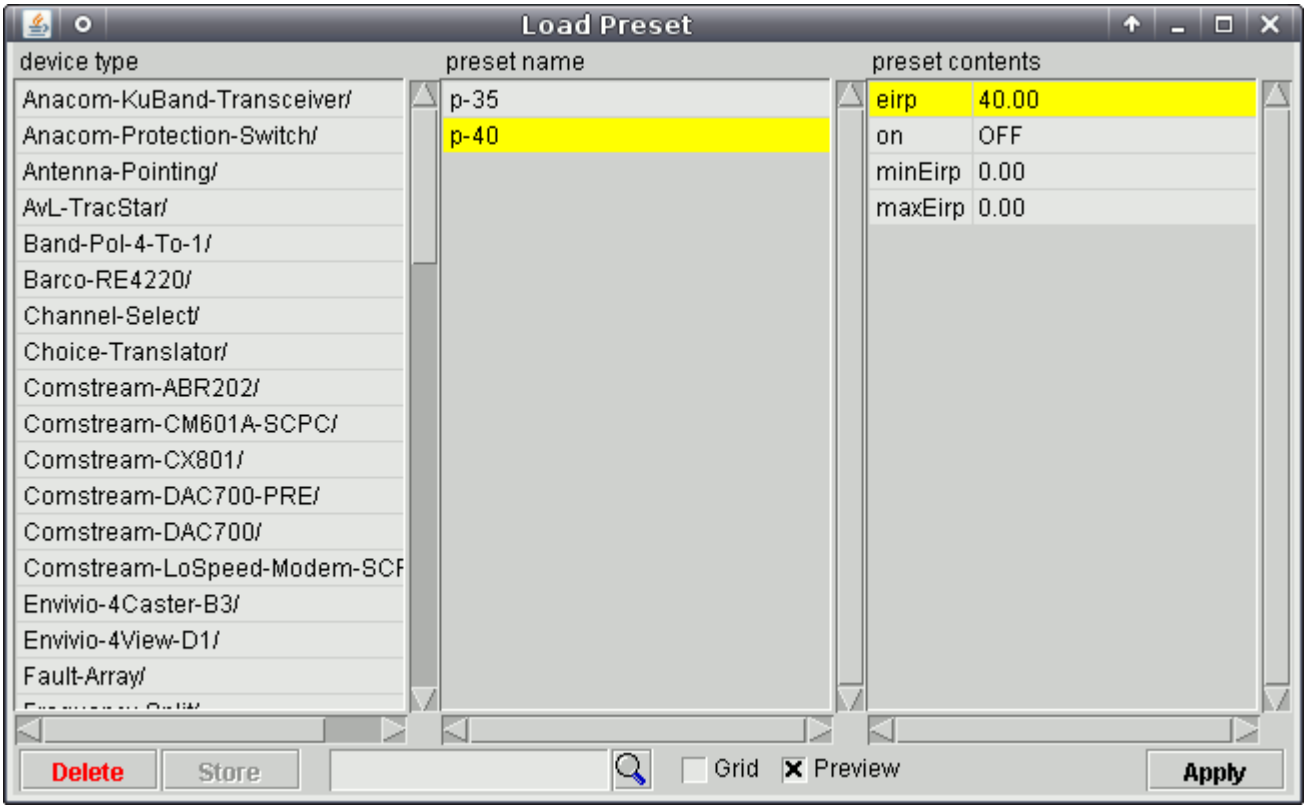
You may add this command as a start menu entry or as a desktop icon to your working environment.

3.7 Managing Device Presets

Most of the sat-nms device drivers allow to store a snapshot of the device's settings in 'device preset' files and let the operator restore these settings at a later time. The load / store preset feature is accessible though the tool bar of the device's ' Device Window '.

Presets are stored at the server your client program connects to. In a M&C environment this is the M&C server, with a NMS environment this is the central NMS server. This means in the latter case a preset may be stored while operating one VLC and later may be applied to a device on another VLC.

Presets always are dedicated to the device (driver) type they were recorded with. This is because is makes no sense to apply the settings of a beacon receiver to a DVB encoder. The target device will not 'know' what to do with the receiver settings. Even with similar devices (e.g. two IRDs from different vendors) there may be differences between the number and type of settings the devices understand. Hence, the software stores the preset files in a "by device driver" structure.



To load a stored preset, click the 'load preset' icon in the device window tool bar and select the preset in the middle list of the displayed dialog. If the *Preview* mark at the bottom of the dialog is checked, the right column shows the selected preset's contents. You may want to uncheck the *Preview* mark if your your connection to the M&C/NMS Server has a very poor bandwidth. Press *Apply* to apply the settings stored in the preset and close the dialog.

If there is a huge number of presets defined for the actual device type, the search function helps to find the proper preset file: Enter a search key into the field at the dialog's bottom line and press the search button to display only presets which contain the key in their name (the search is not case sensitive). You may specify multiple keys, separated by space characters. Only presets which contain *all* of the keys in their name will appear in the list.

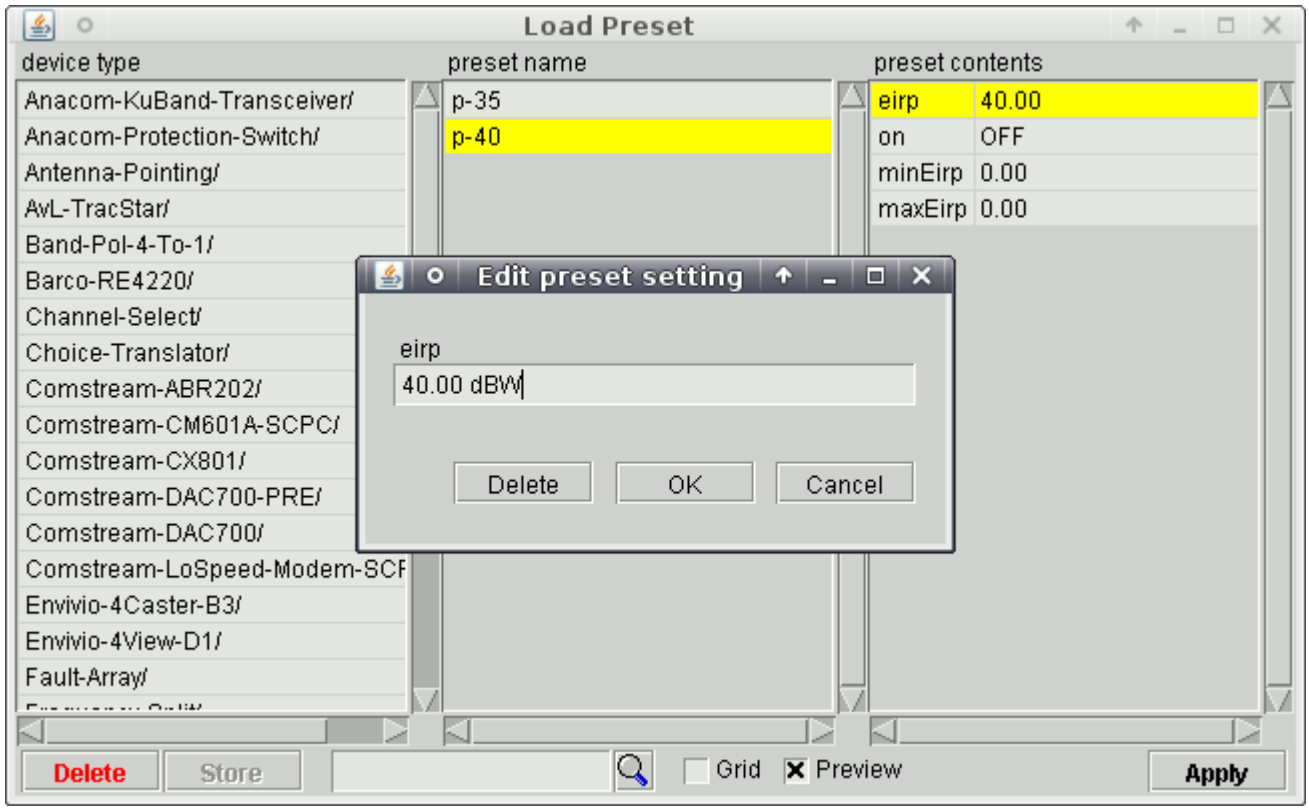
You may align the list of preset names to columns along the dashes in the names by checking the *Grid* mark. This specially is useful, if you strictly follow a certain naming scheme for your presets (e.g. something like 'satellite-channel-variant').

As mentioned above, presets are dedicated to the device type used to record them. However, for special applications the preset load dialog allows to apply a preset from another device type. For this, select the device type in the left column and then the preset file in the middle column. The program displays a prompt you have to quit, to prevent you from accidentally applying a preset from a wrong device type.

Editing Presets

If your system administrator gave you the permission for this, you are able to modify stored presets. The software supports both, permanently modifying a preset definition but also

changing some parameter temporarely before applying the preset:



Be sure to have the *Preview* mark checked and select the preset to modify. Double-clicking a parameter in the right column opens a small *Edit preset setting* dialog showing the value stored for this parameter. You may change the value to your needs and quit the dialog with the *OK* button to accept the modification. Clicking *Delete* removes the selected parameter from the preset which means that this parameter remains unchanged when the preset gets applied. *Cancel* leaves the dialog without changing the parameter.

Changes made to the preset file this way are temporary, they are made in a local copy of the preset. If click to *Apply* , the modified preset gets applied to the device. To store the edited preset permanently at the server, click the *Store* button which is now activated since the preset has been modified.

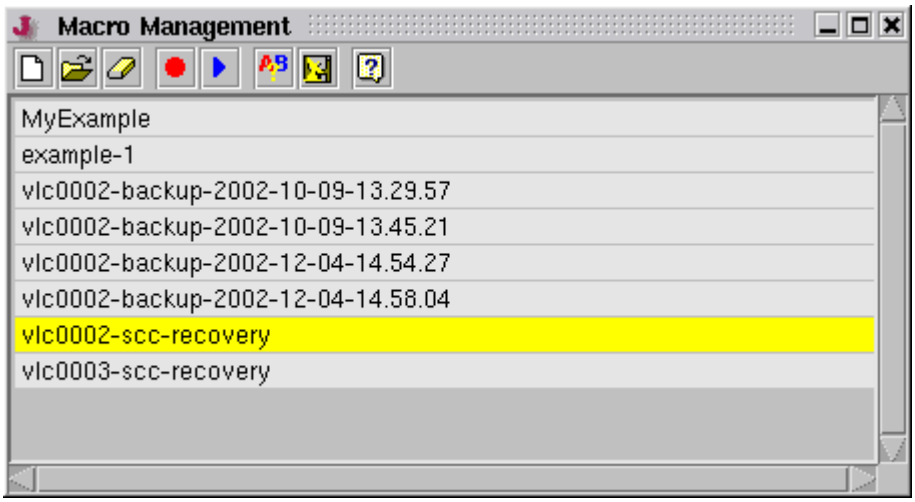
3.8 Recording and Playing Macros

The sat-nms software includes a powerful macro recording and playback feature which lets you automate complex equipment settings to one mouse click. The following chapters describe the user interface windows which give access to this feature.

3.8.1 The Macro Management window

The Macro Management window is the starting point for all operations like recording, playing, editing or deleting parameter setting macros. The window is shown if you click to the ▶ button in the tool bar of the M&C / VLC main window. To get an introduction to parameter setting

macros, see the chapter ' Concepts ' above.



The window shows a list of all macro definitions stored at the server, double clicking a list item opens the macro in the Macro Edit window . Other functions may be called using the tool bar buttons as explained below.

- --- Opens the Macro Edit window with a new, empty macro definition.
- --- Opens the selected macro in the Macro Edit window .
- --- Deletes the selected macro.
- --- Starts recording a new macro definition.
- --- Plays the selected macro.
- --- Compares the selected macro to another one. Please note the macro comparison function ignores all macro directives and does no parameter substitution. This function primary is meant to compare macros which have been recorded with the backup function contained in the SYSTEM device.
- --- Saves the selected macro as the SCC recovery macro to the actual VLC. This icon only is active if the Macro Management window has been called in the context of an SCC enabled NMS application.

Remark

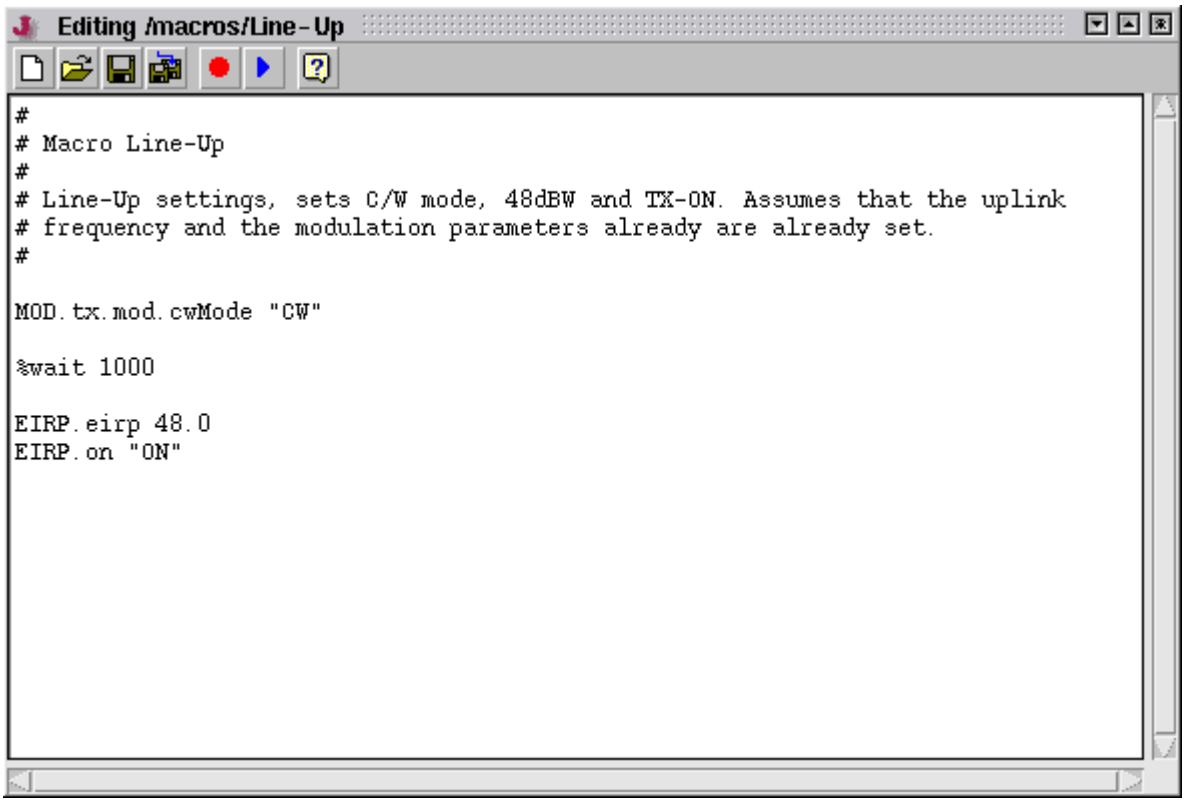
In an NMS, macros normally only can be played while the VLC this Macro Management window refers to is 'online'. An exception from this is a NMS controlling it's VLCs via SCC. Here macros also may be played for 'offline' VLCs, the commands get streamed to the SCC outbound communication channel in this case.

Macros played through the SCC outbound channel may be used to reinitialize a VLC's SCC transmit equipment if any attempts to setup a point to point connection to this VLC fail.

3.8.2 The Macro Edit window

The Macro Edit window lets you edit a macro definition (it's just a plain text file), play the edited macro or append additional steps using the record function. Before you start to edit your first macro, you might want to read the chapter ' Concepts ' at the beginning of this section

which explains the theory, file syntax and usage of these macros.



Handling the editor window (specially using cut & paste) slightly differs for Windows and Linux based computers. The window uses the cup & paste scheme of the operating system, so CTRL-C/CTRL-V works only on Windows based computers while pasting with the middle mouse button only is supported by Linux based machines. Below the tool bar functions are explained:

- --- Clears the edit buffer, lets you edit a new macro definition.
- --- Loads an existing macro definition into the editor.
- --- Saves the edited macro.
- --- Saves the macro using a different name.
- --- Starts a recording session which appends steps to the macro you're actually editing.
- --- Plays the macro in the edit buffer. The macro is played directly from memory, not saved.

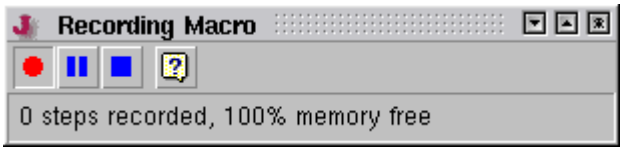
Remark

In an NMS, macros normally only can be played while the VLC this Macro Management window refers to is 'online'. An exception from this is a NMS controlling it's VLCs via SCC. Here macros also may be played for 'offline' VLCs, the commands get streamed to the SCC outbound communication channel in this case.

Macros played through the SCC outbound channel may be used to reinitialize a VLC's SCC transmit equipment if any attempts to setup a point to point connection to this VLC fail.

3.8.3 The Macro Record window

The Macro Record window lets you control the recording of a parameter setting macro. It is shown whenever a macro is recorded. Closing this window in the title bar of the window abandons the recording session, any recorded steps are lost.



The window shows the number of steps which have been recorded in the actual session and the percentage of available memory the macro actually occupies. The recording process is controlled by the tool bar buttons:

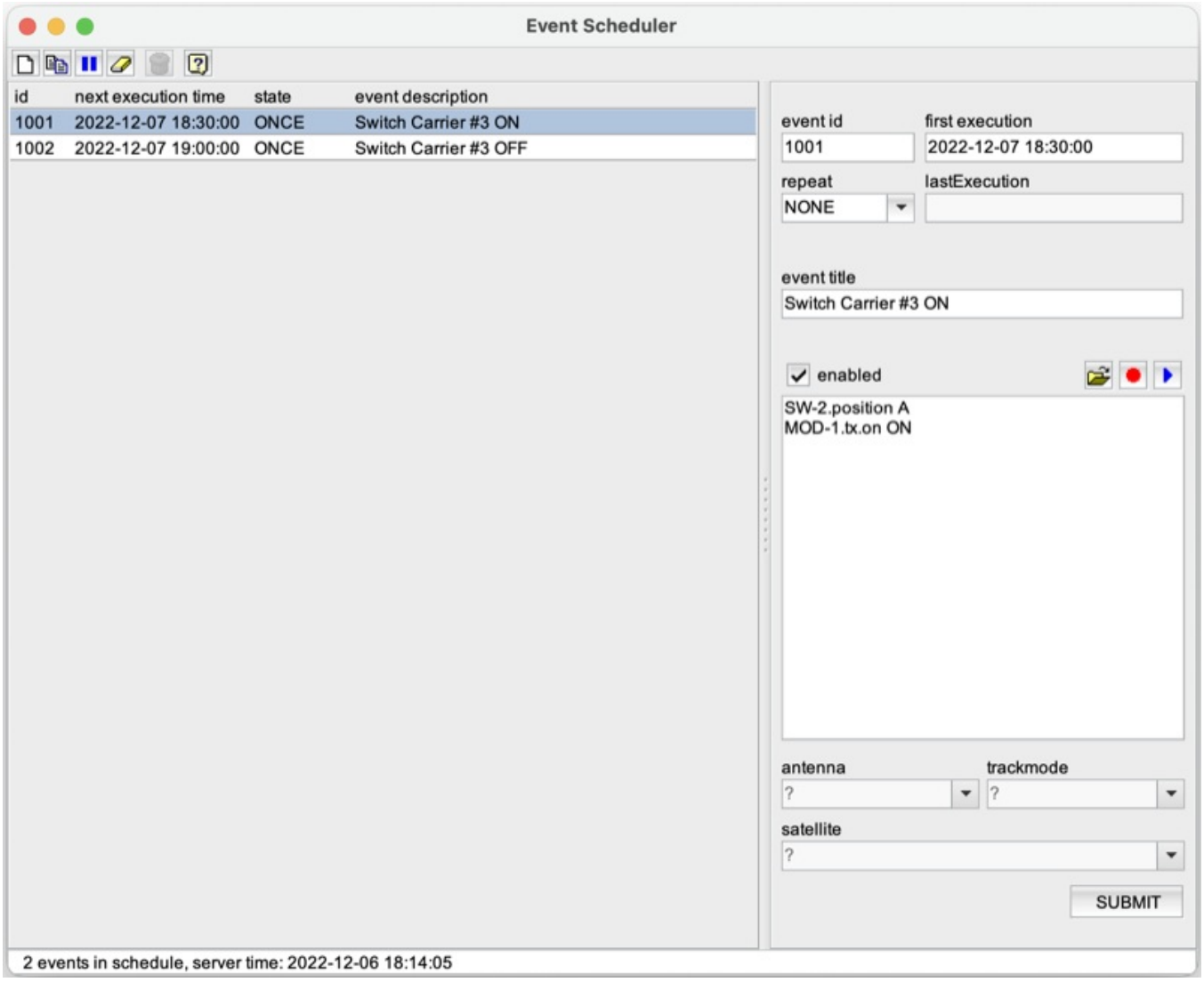
- --- If pressed down, recording is active. This is the default after the window appears. You may pause recording by pressing the pause button described below. To resume recording, click to the record button.
- --- Pauses recording. If, e.g., you find during a recording session that some device settings are not set as you expect, pause recording, set these settings and then resume recording the macro.
- --- Stops the macro recording session and loads the macro into the Macro Edit window. You may modify the macro there, add a comment and store it with an arbitrary name.

3.9 The Macro Scheduler

The macro scheduler included in the sat-nms M&C software lets you execute macros at planned times. You access the macro scheduler from the M&C main window by clicking to the toolbar icon which shows a clock icon.

The scheduler supports events which are executed once at a given time as well as repeated events on a daily, weekly or monthly base. Repeated events are executed in the planned was up to a given end date. Monthly repeated events are executed every month at the programmed day of month. If you program a monthly event for 29.-31. of a month, it will be executed at the last day of a month which is shorter.

The macro scheduler window shows a scrollable list of planned events and provides controls to add, edit or delete events. Like with the macro management window, you may either record the macro you want to be played at the event's time or enter the commands manually. It is also possible to load a stored macro into the event.



The left side of the window shows the list of scheduled events, the right side shows the properties of the actually selected event (or the properties of the new event if you are about to create a new event). Events are sorted by the time of their next execution, the event to be executed next appears on top of the list.

By default the scheduler deletes events once the last execution of this event has been done, however, the scheduler may be configured to keep these events. In the latter case, outdated events appear at the bottom of the list, grayed out with the state DONE.

The toolbar at the top of the window contains all functions to manage the schedule, to add or remove event or to toggle their enabled state:

- Clears the properties section on the right, lets you enter the properties of a new event.
- Makes a copy of the actually selected event into the properties section on the right. You may edit this copy to your needs, the edited event will be added to the schedule when you click "Submit".

- Toggles the enabled state of the actually selected event. Disabled events appear grayed out in the event list, the macro execution is suppressed for a disabled event.
- Deletes the actually selected event from the schedule.
- If the scheduler is configured to keep outdated events in the schedule and if there are such events in the schedule, this toobar button becomes active. Clicking it cleans up the schedule an deletes all outdated events.
- Shows this help page.

The section of the right side of the window contains the properties of the actually selected event of a new event you are about to create. Below a short description of all entry fields and buttons on this section:

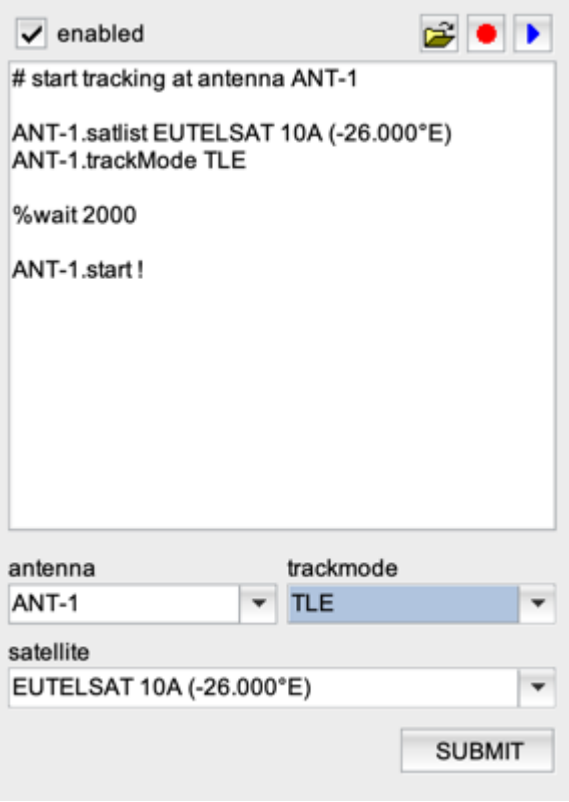
- **event id:** The event id is a unique identifier issued by he scheduler for each event. The event id cannot be changed by the operator.
- **first execution:** The date and time of the first execution of the event (for repeated events) or simply the time of execution for an event executed only once. Be sure to enter the date/time exactly in ISO format: YY-mm-dd HH:MM:SS, the program will reject entries formatted in another way. Double click into the entry field to open a pop up calendar widget which assists you entering the date.
- **repeat:** The repeat mode for the event. This may be one of:
  - **NONE:** The event is executed once at the date and time specified in "first execution".
  - **DAILY:** The event is executed first at the date and time specified in "first execution", then at all following days at the same time but not beyond the day given in "last execution" below.
  - **WEEKLY:** The event is executed first at the date and time specified in "first execution", then every seven days at the same time but not beyond the day given in "last execution" below.
  - **MONTHLY:** The event is executed first at the date and time specified in "first execution", then every month at the same day of month and time but not beyond the day given in "last execution" below. If you program a monthly event for 29.-31. of a month, it will be executed at the very last day of any month which is shorter.
- **last execution:** The day of the last execution of the event.
- **event title:** A descriptive title for the event which will appear in the event list at the left side.
- **enabled:** If checked, the event is enabled. An enabled event will execute its macro at due time, a disabled event will skip this.
- **macro text:** The large field below the "enabled" check mark contains the macro text to be executed. The format and syntax of the macro text follows exactly the definition given in section Automating Procedures With Macros . Unlike the standard macro facility of the software which stores macros as text files in a separate directory, macros of the scheduler are store with the schedule itself, they are not accessible from the software's macro player.
  - Lets you load an existing macro file into the macro for this event. The loaded file will replace the previous content of the entry field.
  - Lets you record the macro. While recording is active, all parameter settings you make at the GUI are recorded an added to the macro.

- ▶ By clicking this button, you can immediately execute the edited or recorded macro for testing.

- **SUBMIT:** Clicking SUBMIT actually adds the new / updates the edited event at the schedule. The program issues a query before executing this.

Antenna Scheduler

The Macro Scheduler provies a function to create easily a macro to start tracking at avertain antenna. For this function there are a number of selection fields at the lower right corner of the window, just above the SUBMIT button.



If you change any of the selections *antenna*, *trackmode* or *satellite*, the software replaces the the macro text with a fragment which sets up the antenna for the selected satellite and tracking mode and starts tracking on it. Additional commands / parameter settings may be aded to the macro before the schedule event gets saved by a click do SUBMIT.

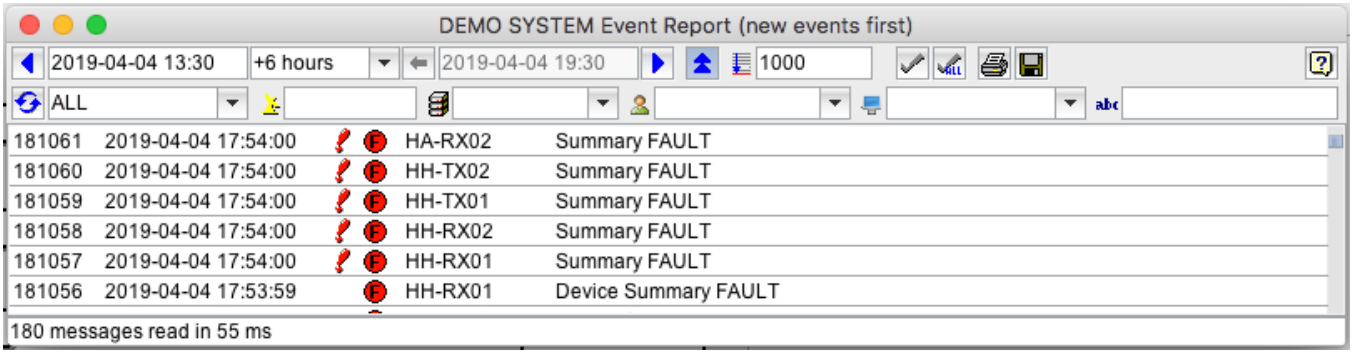
The antenna scheduler function is only available for antennas which are managed by Antenna-Management devices. If there are no such devices configured in the M&C, the fields assigned to the antenna scheduler function apear disabled.

3.10 The Event Report Window

The Event Report window lets you generate filtered reports from the event database maintained by the NMS. This database contains all event (fault) messages issued by the VLCs during the last months (the exact depth of event storage is configurable during system

installation).

The event report shown in the window gets automatically updated when you change the time range to display or the sort order. Changing the filter options in the second line of the toolbar does not automatically update the report, you have to click the reload button or press ENTER in one of the filter fields for this. This lets you combine various filter specifications without being interrupted by the generation of intermediate reports.



Data shown in the list columns

- *column --- description*
- 1 --- The event ID. This is a unique number the software assigns to each event which arrives at the NMS.
- 2 --- The time stamp when the event was originated. The originating VLC sets this, so the time relies on the accuracy of the VLC's clock.
- 3 --- The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- 4 --- The message priority. The message priority decides on the urgency, the message gets delivered from the VLC to the NMS. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - --- This is an informational message (lowest priority)
  - --- This is a fault message (medium priority)
  - --- This is an alarm message (highest priority, the VLC tries to deliver this immediately, regardless of the dial timing settings)
- 5 --- The primary message source. This usually is the ID of the VLC which issued the message. Messages marked with 'USER' or 'NMS' in this column have been generated locally in the NMS.
- 6 --- The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a VLC which caused the event.
- 7 --- The message text.

Tool-bar Functions

On top of the event report window, there is a 2-line toolbar which lets you control all aspects of what the event report shall contain.

The controls in the upper line between the two blue arrows define the time range to be displayed in the report. For this, the event report windows knows two different modes. One lets you define the time range as start time plus a duration picked from a list of commonly used values. The second mode lets you specify start and end time explicitly. The small button with the horizontal gray arrow symbol lets you switch between the modes. The gray arrow always points to the parameter which is actually active.

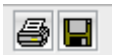
The blue arrow buttons enclosing this section of the tool bar let you increase or decrease the start point in increments of the selected duration.

A double click to the start or end time fields opens a calendar dialog window.

The button with the double blue vertical arrows on it toggles the order in which the events are shown. With the arrows down, the newest event is at the bottom of the list, when pointing up the list is ordered from newest to oldest. The default sort order is determined from a parameter set in the "client.properties" file located at the NMS or M&C server. Setting the key "gui.eventSortOrder" to "true" or "false" defines the default behavior of the event report window.

The entry field right of the sort order button defines the maximum number of events one report may contain. There is a default value for this (1000 in most cases), which is a good trade off between performance and the number of events displayed at a time. You may change this limit to your needs, but be aware: reports with millions of events may take some time to be collected.

These buttons lets acknowledge either the selected fault message only or all pending fault messages contained in the event log. Acknowledging a message is always logged as an additional event.



The 'print' and 'save' buttons are quite self-explanatory. 'print' sends the actual report to the printer using the system print dialog of your client computer. Depending on the operating system you are using, you probably have the possibility to preview the print and to select the pages of the report you actually want to be printed.

The 'save' button lets you save the actual report as a CSV file on your client computer. A file

select dialog box pops up, you have to enter or select a file name and click 'save' in order to have the file actually stored. The columns of the CSV file are separated by semicolons, you may import this file for example into a spread sheet program of your choice.

The reload button forces a report to be generated once more, for example to include some new events which may have ocurred since the time when the report was created. During a long lasting query, the button turns into a red 'stop' button, you may click to this to abort a pending query.

The priority selector defines the priority (or type) an event must have to be listed in the report. The selection *ALL* displays all events. *FAULT* includes events which at least have fault priority. *ALARM* displays alarm messages only (see the description of alarm priorities above). The selection *NOT ACK* shows all messages which (still) need to be acknowledged.

This filter lets you filter messages along their primary source. The primary source of a message is the VLC in an NMS environment. For stand alone M&C systems this filter is actually not applicable.

Any non blank value in this field filters those events which have their primary source field (usually the originating VLC) set to this value. If you enter multiple words in this field, all messages are shown which contain *all* words in their primary source at an arbitrary position. The filter is not case sensitive.

If the first character in this field is '-', the filter gets inverted. Example: '-vlc0003' lists all events which do *not* come from vlc0003.

This filter lets you filter messages along their secondary source. The secondary source is usually the name of the device which caused the message. With M&C systems, the entry field for the filter is a combo-box suggesting all device names used in the system in a drop down list. You may enter free text anyhow.

Any non blank value in this field filters those events which have their secondary source field set to this value. If you enter multiple words in this field, all messages are shown which contain *all* words in their secondary source at an arbitrary position. The filter is not case sensitive.

If the first character in this field is '-', the filter gets inverted.

The user name filter lets you restrict the report to messages originated by a certain user. Login messages and messages logging the active change of a parameter are stored with the name of the user who is responsible for this event.

The user name filter is a combo box which lets you either type the name of the user or selected from a drop down list with all known user names at this system. The filter is not case sensitive. You may type only a part of the user name, the filter shows all messages containing this character sequence in their user name field.

Please note, that this filter - if not blank - suppresses all messages which have no user name stored. No faults, no warnings are displayed with this filter.

The client host name filter lets you restrict the report to messages originated from a certain client PC. Login messages and messages logging the active change of a parameter are stored with the origin (hostname or IP4 address) of the event.

The client host name filter is a combo box which lets you either type the name of the client computer or select it from a drop down list with all known client names at this system. The filter is not case sensitive. You may type only a part of the client host name, the filter shows all messages containing this character sequence in their host name field.

Please note, that this filter - if not blank - suppresses all messages which have no client host name stored. No faults, no warnings are displayed with this filter.

The last filter is applied to the message text. Any non blank value in this field filters those events which contain this value in their message text (partial match). If you enter multiple words in this field, all messages are shown which contain *all* words in their message text at an arbitrary position. "one two" as a message filter will match for example the message text "received two ones". The filter is not case sensitive.

If the first character in this field is '-', the filter gets inverted. Example: '-OK' lists all events which do *not* contain OK in the message text.

Remarks

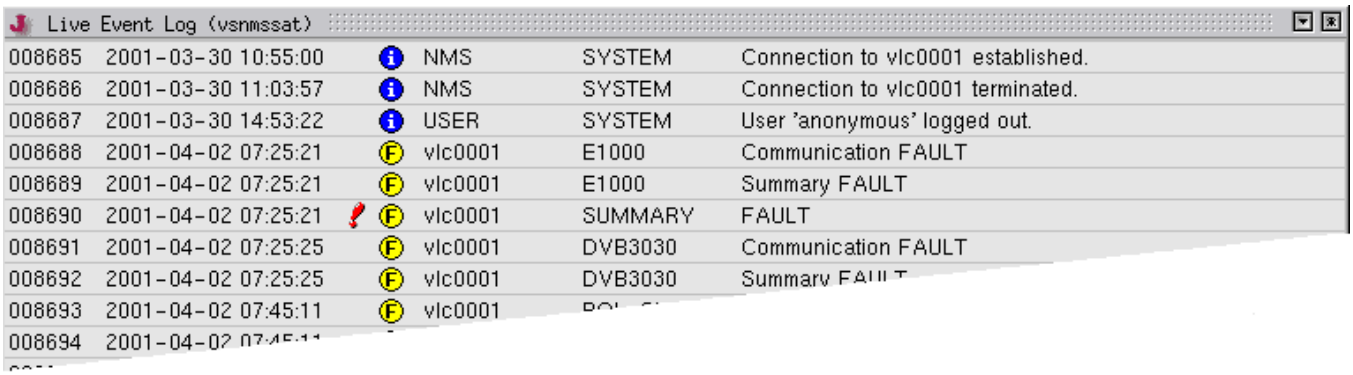
- The messages appear sorted by the time when they were originated.
- If you click to a message in the report, the time when the message was received at the NMS is displayed in the status line at the bottom of the window.
- With the right mouse button you get a context menu on each event, providing additional functions depending on the event's contents.
- The filter functions are combined using a logical AND. The report contains those events of the selected time range which match all of the filters.
- A new report is generated with the actual filter settings if you press ENTER in one of the entry fields or if you click to the reload button.

3.11 The Live Event Log Window

The Live Event Log window shows the recent event messages as they are received from one of

the VLCs. Unlike the Event Report window , which sorts the events by the when they were originated, the Live Event Log window shows the messages in the order of their arrival at the NMS. The window has a fixed size and always shows the last 25 messages.

The sort order of the event list (newest first / oldest first) may be toggled using the context menu shown with a right mouse button click on any line of the list. The default sort order is determined from a parameter set in the "client.properties" file located at the NMS or M&C server. Setting the key "gui.eventSortOrder" to "true" or "false" defines the default behavior of the live event log window.



Data shown in the list columns

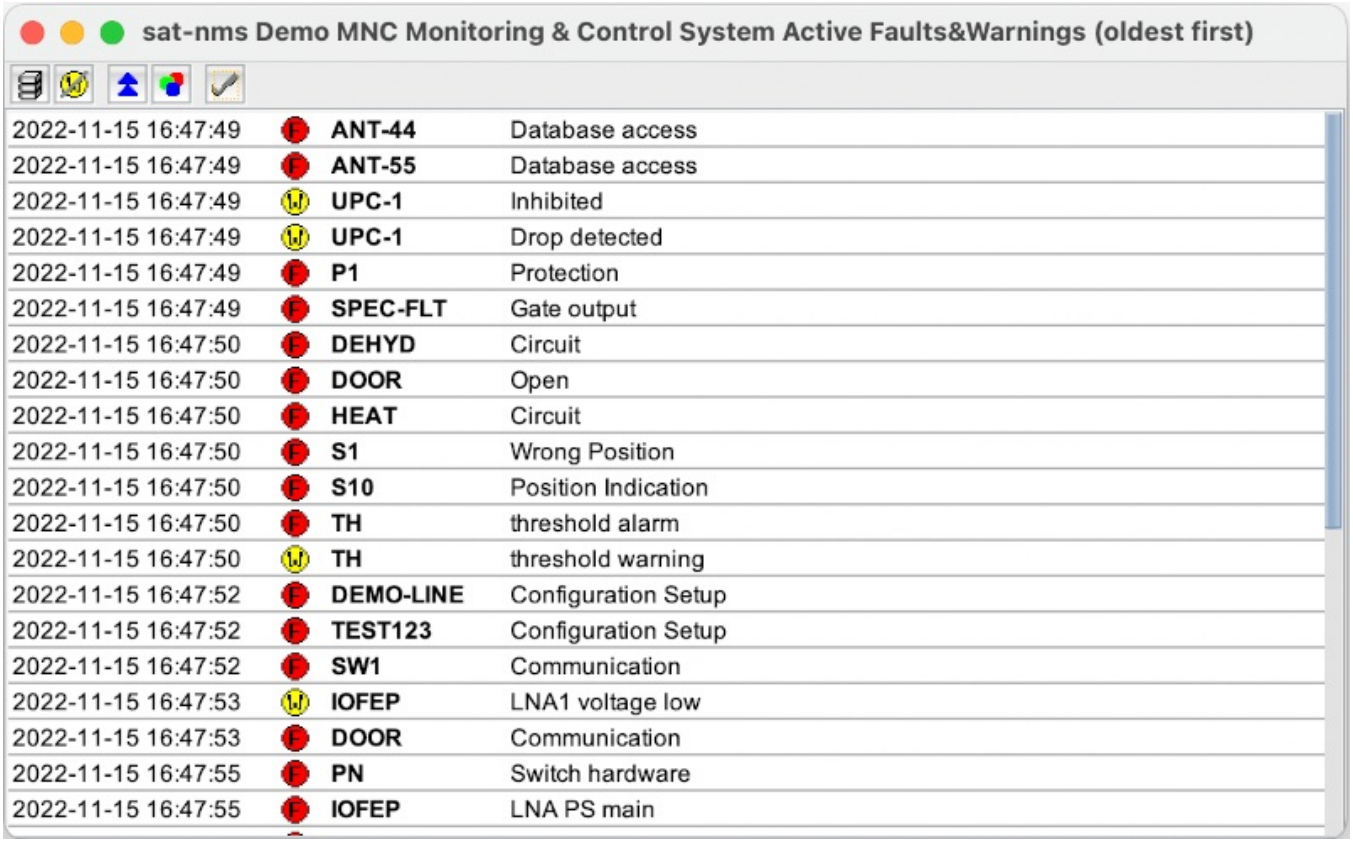
- Column 1 --- The event ID. This is a unique number the software assigns to each event which arrives at the NMS.
- Column 2 --- The time stamp when the event was originated. The originating VLC sets this, so the time relies on the accuracy of the VLC's clock.
- Column 3 --- The 'need acknowledgment' mark. Messages showing a red exclamation mark in this column are fault messages which need to be acknowledged by the operator.
- Column 4 --- The message priority. The message priority decides on the urgency, the message gets delivered from the VLC to the NMS. It also is a search criteria in the Event Report window. The following priority symbols may appear:
  - --- This is an informational message (lowest priority)
  - --- This is a fault message (medium priority)
  - --- This is an alarm message (highest priority, the VLC tries to deliver this immediately, regardless of the dial timing settings)
- Column 5 --- The primary message source. This usually is the ID of the VLC which issued the message. Messages marked with 'USER' or 'NMS' in this column have been generated locally in the NMS.
- Column 6 --- The secondary message source. This gives a more detailed information about the source of the message. It may be the name of the device in a VLC which caused the event.
- Column 7 --- The message text.

You may acknowledge messages marked with the red exclamation mark by double-clicking to the message line. This makes the exclamation mark disappear, an acknowledgment message gets added to the end of the log.

Please note, that if you close the Live Event Log window and open it again, any acknowledged message will appear with the exclamation mark set again. This is because the recognition of acknowledged messages depends on message sorting which does not happen in this window.

3.12 The Active Faults Window

The Active Fault Windows lists all actually pending faults and warnings in the VLC / M&C. The window automatically updates its content as the faults appear or disappear.



The toolbar buttons of the window let you select how the window displays the list of active faults:

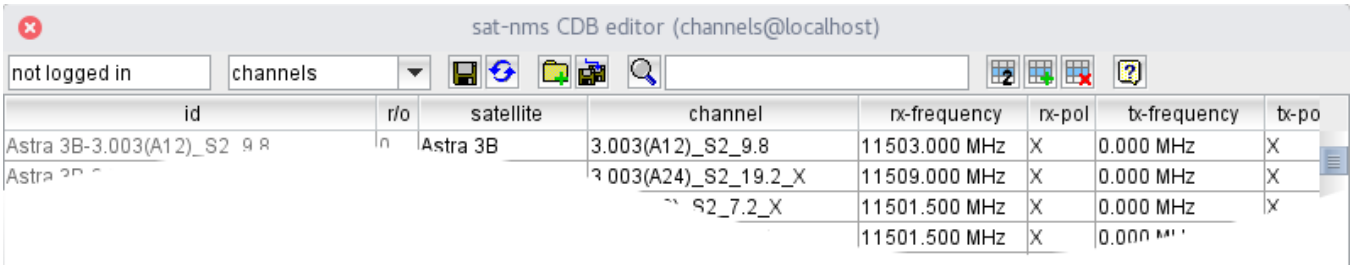
	If pressed, the faults are grouped by device name, if not they appear sorted by date.
	If pressed, the window suppresses al warnings, shows faults only. If not, faults and warnings are displayed.
	If pressed, newest events are shown on top. If not, oldest events are shown first.
	If pressed, the background of each line gets colored red for faults, yellow for warnings.

✔	Makes the "new fault" mark in the main window disappear and silences the beeper.

3.13 CDB Central Database Table Editor

The text based channel database (CDB) is a legacy tool to mangage satellite channel rx-/tx-parameter in a conveient way. It has been replaced with a SQL based database solution which entirely is operated through the sat-nms Web client. The description below is contained in the manual for reference only, it may apply to older M&C installations using the sat-nms Java client.

The sat-nms CDB table editor is a versatile tool to manage satellite channel lists in an easy and clear way. The table editor displays the contents of the CDB in spreadsheed like layout, lets you edit, add or delete channel definitions. Beside this, the table editor may be used to export or import the CDB table or parts of it on your client PC.



The picture above shows the table editor's toolbar. Below you find a short reference for all functions which are accessible here. A more elaborate description of the table editor's capabilities you find at the paragraphs forther down at this page.

- --- The login field. To log in, click into the field and enter your user name / password in the dialog popping up. You need to be logged in to save the edited table back to the server, all other functions work without login als well.
- --- The database selection. The M&C server may maintain other database tables beside the satellite channel list. Using this drop down selection you may switch between these databases. With your sat-nms installation, the field may be missing or limited to the 'channels' selection only.
- --- Saves the edited table back to the server. You need to be logged in for this.
- --- Reloads the table from the server. This abadons all changes since you opened the editor (or since the last save).
- --- Opens the import dialog. Lets you read a CSV file with channel definitions from you local PC and import these definitions into the CDB.
- --- Opens the export dialog. Lets you export all channels (or only the selected ones) to a CSV file on you client PC.
- --- Toggles the search function. Requires a search pattern to be entered in the field at its right.
- --- The search pattern field. Enter a search pattern here to select all channels matching this pattern. See 'Searching channels' below for a more detailed description how seaching works.

- --- Duplicate row. Makes a duplicate of the (topmost) selected channel.
- --- Add row. Inserts new channel definition above the (topmost) selected channel. The new channel gets initialized with default settings.
- --- Delete row(s). Deletes the selected channel or all selected channels if more than one is selected.
- --- Shows this help page.

Document style workflow

The table editor handles the CDB table like a document. When you start up the editor, it requests the complete database contents from the server and presents it in the editor window. It does not change the database until you click the save button to write back the changed channel list to the server. At this point all devices using the channel table will be informed about the changed database.

This workflow implies two things:

1. You can safely play around with the editor, change, delete or add channel definitions without disturbing the normal operation of the station. At any point you can abadon the editing process without changing the original data.
2. If two persons edit the database using two instances of the table editor at the same time, only the changes saved lastly will be taken by the CDB. This is the same as two persons edit the same document on a shared drive: The one wins who saves the changed document al last.

Logging in

A login as a user with a privilege level 100 or higher is required to write a modified table back to the server. All other functions, viewing channels, exporting them etc. are possible without login.

Login is done quite similar to the sat-nms M&C software, simply click to the login field in the upper left corner of the window and enter your user name and password at the dialog popping up.

Selecting the database

Some software installations may use multiple CDB databases, e.g. one for satellite channels and another for BISS keys. These installations show a selection field for the database to edit beside the login field. This field may be missing with your software installation.

Sorting the channels

The table editor normally presents the channels sorted alphabetically by the channel ID. By clicking to a column header of the table, you may sort the table by this column, e.g. by receive frequency. Clicking again to the table header reverts the sort order. Clicking to the header of the first column (channel ID) restores the original sort order.

Searching channels

When entering a text in the search field of the tool bar, the editor highlights all channels which contain this text in their channel ID. Beside this more complex searches are possible:

- Multiple words separated by space characters are searched and combined with a logical AND: "astra S2" for example marks all channel containing "astra" AND "S2" in their channel ID.
- Search words may be prepended by the name of a table column (as shown on the table header) followed by a colon. This makes the program search this word in this column. Example "EUT07 fec:7/8" marks all channels which contain "EUT07" in their channel ID and have a fec "7/8" set.
- The search is not case sensitive. The 'Search' button beside the search field toggles the selection

Selecting channels

A single channel is selected by clicking somewhere into the table row representing this channel. Holding down the SHIFT key while clicking to another table row selects a range of channels between these rows. Holding down the CTRL key while clicking to some channels adds these channels to the selection (or removes them if they are already selected).

Editing channels

You may change any channel parameter 'in place' in the table. For selection parameters simply click to the parameter field and select the new value from the drop down list. For numeric or textual parameter you need to double click the table field to switch it to edit mode. As soon as one value has been changed, the "MODIFIED" indicator shows up at the upper right corner of the window.

When you change the satellite name or the channel name, the program makes a new channel identifier from this and checks if this identifier is unique in the channel list. If not, a suffix is added to the channel name to make the identifier unique and a warning ist displayed.

If you are using the 'Import EBU channels' at the RX-Channel device, the channels imported from the EBU channel list are marked read-only anf cannot be edited in the table editor.

Adding new channels

There are two options to add a new channel to the database. 'Add a channel' means to add a new channel named 'NEW' and filled with defaults for all parameters. The second way to add a new channel is 'Duplicate channel'. As the name suggests, this duplicates the selected channel. All parameters are copied from the source channel and a suffix is added to the channel name to make the channel ID of the new channel unique. The new channel gets added above the actually selected row. If multiple rows are selected, the new channel is added above the topmost selected channel, this also acts as the template for the 'duplicate' function

Deleting channels

To delete one or more channels from the table, select the channels to delete and then click the 'delete row' button in the tool bar.

Importing channels

The table editor permits to import a set of channels from a CSV file which is located at your client PC. To import a file, click the 'import' button in the tool bar, select the file in the dialog box popping up and finally click 'Open' in this dialog box.

The table editor imports the file and adds the database records defined in this file to the table. It does this in an intelligent way:

- If an imported record has a duplicate in the table which matches the imported one for all parameters exactly, the imported channel definition is discarded. No duplicate is created in this case.
- If the imported channel has a duplicate in the table with the same channel ID but with different parameters, the channel gets imported and the channel name gets a '_DUP' suffix added.
- All other records in the imported file are simply added to the table.

The file format the import function expects, depends on your software installation and software version. It is always safe to import files which have been created using the export function on a sat-nms M&C with the same version number.

Exporting channels

You may export either all channels in the table or a subset of them to a local CSV file. Select the files to export and click the 'export' button in the tool bar. Then select or enter the file name you want to save to an click 'Save'.

If no channels are selected, the software exports all channels. You may use the search button (the magnifier glass) to remove an existing selection. If only one record is selected, the software will export this record only.

Saving the changed table

The 'save' button writes the changes you made back to the CDB server. You need to be logged in with a privilege level of at least 100 to do this operation. Once you click 'save', all RX-Channel and TX-Channel devices in the M&C get informed about the changed channel list.

If you made changes to the table and close the editor without saving the table, the program shows a warning about this.

Reverting all changes

The table editor offers the 'reload' button in the tool bar to discard all changes ant to load the table again from the CDB server. You may want to use this if you accidentally messed up the channel table.

3.14 The sat-nms ACU Indoor Unit

The ACU indoor unit is a PC system based on the sat-nms M&C technology. Together with an antenna controller and a beacon receiver it builds the sat-nms antenna control system. Main

features of this are:

- Monitoring and control of the concerned equipment.
- Satellite tracking using step track and orbit prediction tracking methods.
- Analysis and graphical presentation of the satellite tracking performance.

Using the sat-nms M&C platform as the basis, the ACU indoor unit also includes a lot of the benefits of the basic software:

- The software is highly modularized. The subsystems interfacing the tracking engine to the outside world may be configured to control third party products instead of the sat-nms outdoor unit or the beacon receiver.
- The user interface is very clear and contains data analysis tools to display in which way the antenna tracked the satellite.
- As with other implementations of the sat-nms M&C Software, the user interface is an platform independent Java program which may by run on any other computer in the LAN than the indoor unit itself.
- The indoor unit provides a event log database with a HTTP based interface to third party software.

The following chapters explain the concepts behind the sat-nms antenna control system and the architecture of specific configurations of the ACU. This is supplemented by a detailed description of all user screens and the meaning of all adjustable parameters.

3.14.1 Concepts

SatService GmbH offers sat-nms ACU Indoor Unit in two general configurations:

- The first configuration is optimized to work together with the sat-nms ACU/ODM. It utilizes all features of this device and adds a comfortable user interface with some handy utilities to it. The basic tracking functions are handled by the ODM in this case.The chapter 'sat-nms ACU Indoor Unit & Outdoor Module Configuration' describes the functionality and architecture of this configuration, chapter 'The ACU Main Window (with sat-nms ODM)' the appendant user interface. Furthermore you find a full description of the sat-nms ACU / ODM in the document 'sat-nms ACU Outdoor Module / User Manual'.
- The second configuration is directed to applications where the antenna pointing controller is a third party product. In this configuration the ODM / pointing controller is solely used to position the the antenna. All tracking algorithms are performed by ACU indoor unit software.Chapter 'sat-nms ACU Indoor Unit & 3rd Party Controller Configuration' describes the functionality and architecture of this second configuration, chapter 'The ACU Main Window (with 3rd Party Controller)' the appendant user interface. This modularity and flexibility makes the sat-nms ACU Indoor Unit to a universal antenna controller with sophisticated tracking capabilities which can be used with a multitude of antennas. Due to it's configurability it can be deployed together with new antenna installations and as an upgrade for existing antenna systems as well.

3.14.1.1 Tracking Modes

The sat-nms antenna control system -- regardless of the configuration used -- provides the basic types of satellite tracking:

Step Track

In plain step track mode, the ACU performs small test steps with the antenna in regular intervals to see if the receive level becomes better or worse for another antenna position. With the knowledge of the antenna pattern's shape, the tracking engine computes an optimized position from measurements taken at the old position and at the new one. Many parameters such as the step size, averaging times and the overall repetition interval are configurable by the operator.

Adaptive Track

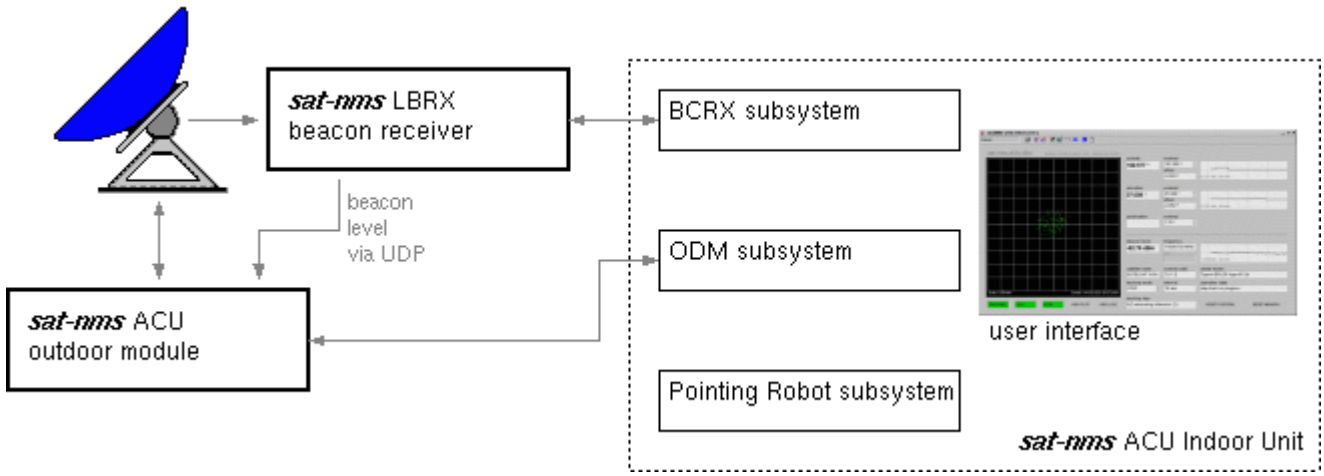
The adaptive tracking mode works much like the plain step track, however the tracking engine in background computes a mathematical model of the antenna motions from the step track results. A great advantage of this mode is that the antenna follows the computed model even if the beacon reception fails. With the plain step track mode, the antenna would freeze in this case until the satellite beacon appears again.

Program Track

With the third tracking mode called program track, the antenna follows a position computed from the satellite's ephemeris data. The tracking engine accepts Keplerian elements in NASA 2-line format or the proprietary Intelsat ephemeris data format for this. For this tracking mode, no beacon reception is required, however the ephemeris data sets must be updated every few days.

3.14.1.2 sat-nms ACU Indoor Unit & Outdoor Module Configuration

The diagram below shows the architecture of a antenna control system entirely built of sat-nms components. The 'sat-nms ACU Outdoor Module' controls the antenna pointing and performs the steptrack and adaptive tracking modes. The 'sat-nms LBRX Beacon Receiver' measures the satellite beacon level and supplies this level value on the Ethernet vie UDP Packets to the ODM. Both devices are controlled by the IDU through an Ethernet network (strictly spoken, the ODM also communicates with the beacon receiver directly, but this is for synchronization purposes only).



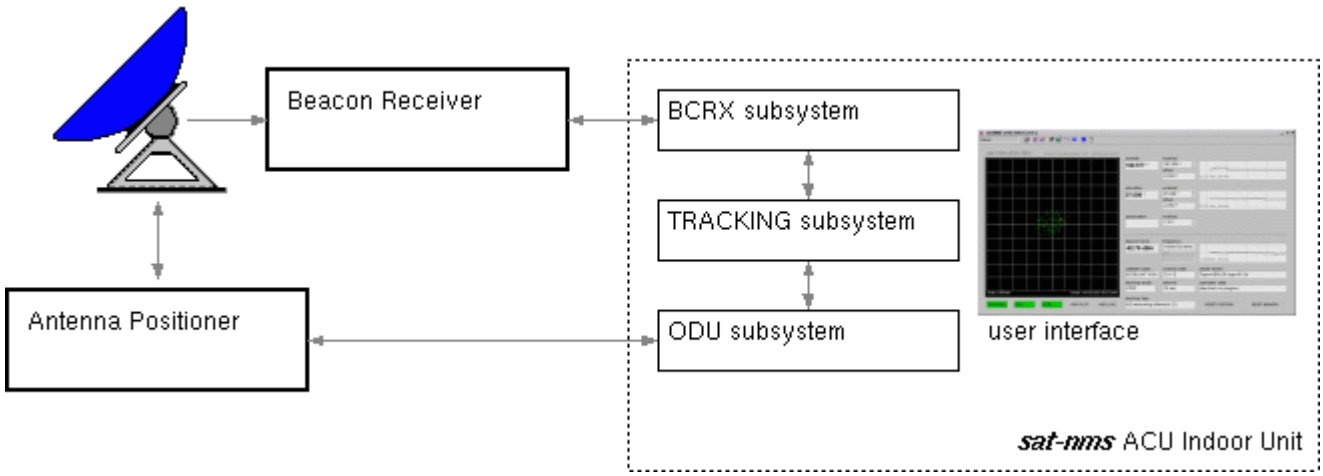
The indoor unit provides a comfortable graphical user interface which permits to monitor and control the whole antenna system. Beside this, it contains the so called 'Pointing Robot' , a software module which controls the antenna pointing according to the data evaluated from satellite ephemeris data sets or read from tables. This versatile tool superseeds the PROGRAM-Track function built into the ODM, which is not used in this configuration.

Configuring the antenna control system in this way is the favored solution. It achieves the best tracking results. More benefits are:

- The components of the antenna control system are optimally coordinated.
- Doing the step track locally in the ODM allows a very close timing of the operations during a steptrack cycle, this makes the tracking less susceptible to atmospheric variations of the beacon level.
- The tracking is completely independent from the indoor unit. If the IDU fails, the ODM still tracks the satellite.

3.14.1.3 sat-nms ACU Indoor Unit & 3rd Party Controller Configuration

The diagram below shows the architecture of a antenna control system using third party components. The 'Antenna Positioner' device solely controls the antenna pointing, it does not perform any satellite tracking. The 'Beacon Receiver' measures the satellite beacon level. Both devices are controlled by the IDU using arbitrary M&C interfaces (serial, TCP/IP). Any tracking functions are performed by a software module in the sat-nms ACU IDU.

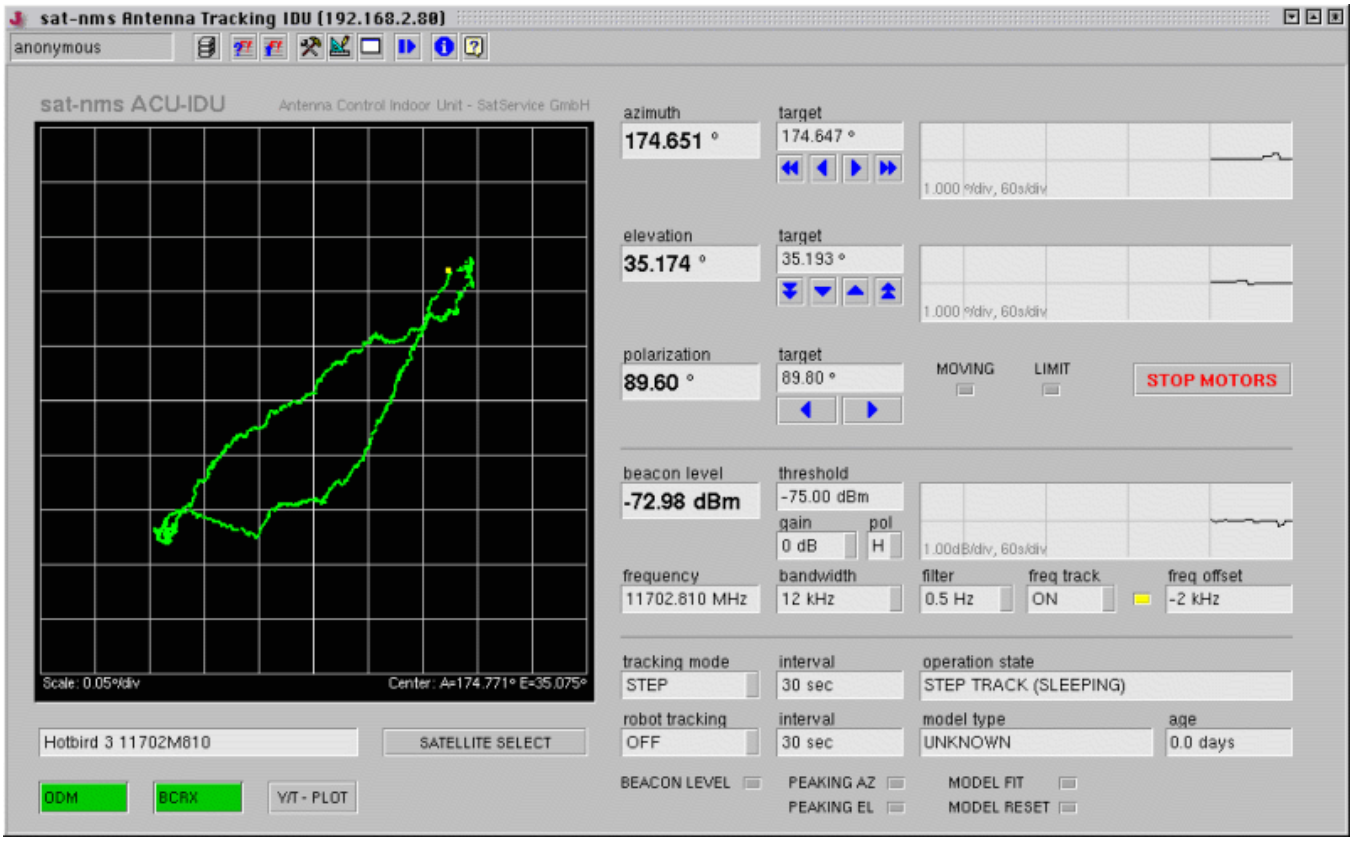


The indoor unit's main task is to perform the satellite tracking in this configuration. The IDU reads the beacon level thru the receiver's M&C interface and directs the antenna to to tracking steps. The tracking algorithms used by the IDU are similar to those used by the ODM, but not exactly the same. This is mainly because there is a longer time between moving the antenna for a test step anf reading out the beacon level.

The main advantage of this configuration is it's flexibility. The IDU is capable to do all tracking modes with any antenna controller which can be remote controlled by the sat-nms M&C system, the same applies for the beacon receiver. The sat-nms ACU indoor unit is a cost effective solution to upgrade existing antenna systems with a modern, network based user interface and a powerful satellite tracing engine.

3.14.2 The ACU Main Window (with sat-nms ODM)

The main screen of the sat-nms ACU Indoor Unit shows the main parameters of the antenna tracking control similar to the front panel of a oscilloscope. The left part of the window contains a diagram which shows the antenna movements of the past 48 hours. The most important settings and status displays are located at the right.



The toolbar on top of the window gives access to user login/logout, the event log and some other functions. As the ACU indoor unit inside principally is a M&C system, the toolbar is exactly the same as for the M&C Main Window .

At the top of the right part of the window thepointing display is located. It shows the actual antenna pointing and a strip chart of the recent history of antenna movements. Below the pointing display you find the beacon level display It reports the actual beacon level in a similar way. At the bottom of the right side the window contains an area with the tracking status display . Finally, below the tracking diagram on the left side, the ACU main screen contains the ACU subsystem menu . These buttons give you access to subsystem screens with all detail parameters of the ACU / tracking.

3.14.2.1 The Pointing Display

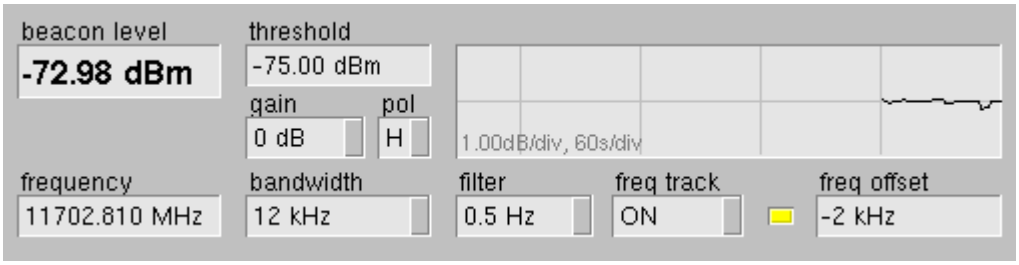
The area in the upper right part of the window shows the actual antenna pointing and provides entry fields to move the antenna to a given position. The example below is taken from an antenna which is not equipped with a polarization motor, the polarization angle display is empty therefore.

The fields labeled 'azimuth', 'elevation' and 'polarization' show the actual antenna pointing as read out from the angle encoders. For azimuth and elevation there is also a strip chart display of the angle. If you choose a high resolution scale for the chart, you can watch the antenna doing the tracking steps in the chart. Click with the right mouse button into the chart to get a pop-up menu for scales and other display options.

The 'target' fields show the actual target values. The arrow buttons below these fields permit to adjust the antenna position in small or larger steps. This only should be done with any tracking switched off to avoid interferences with motions initiated by the tracking engine.

3.14.2.2 The Beacon Level Display

Below the antenna pointing, the ACU shows the actual beacon level. The level value shown is that one read by the ODM from the analog input. This value may differ from the level reported by the beacon receiver directly. There is also a strip chart display of the level. The chart moves with the same speed as the angle charts above, so you directly can watch the influence of tracking steps to the beacon level. A click with the right mouse button into the chart shows a pop-up menu for scales and other display options.



The screen also provides fields to set common parameters of the beacon receiver like the receive frequency and polarization, bandwidth parameters and the frequency tracking option. Changing the values in these fields changes the beacon receiver setting immediately.

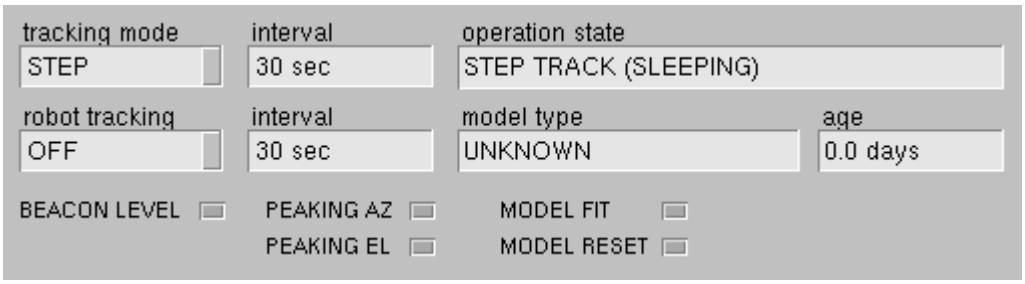
The threshold value contained in this parameter group is the ACU/ODM level threshold rather than the threshold parameter of the beacon receiver itself. The latter is recommended to be

set to a very low value in order to avoid level fault messages from the beacon receiver. The beacon level threshold is to be monitored by the ACU/ODM, hence the level fault can be synchronized with the tracking activities.

The frequency offset display in this group reports the activity of the beacon receiver's frequency track facility, the small yellow lamp is lit while a frequency track cycle is in progress

3.14.2.3 The Tracking Status Display

The lower right area of the ACU main window shows the tracking status display and the basic tracking mode parameters.



The table below gives a short description of each parameter in this area.

- **tracking mode** --- With this parameter you set the basic tracking mode. Available modes are OFF, STEP and ADAPTIVE. This parameter is passed to the ODM.
- **interval** --- With this parameter (upper row) you define in which intervals the ACU shall perform tracking steps. This parameter is passed to the ODM as well.
- **operation state** --- This field shows the general operation state of the tracking engine.
- **robot tracking** --- This switches the pointing robot ON or OFF. If the pointing robot is activated, the tracking mode is forced to OFF anf the antenna of moved on a path evaluated by the robot.
- **interval** --- The interval parameter in the lower row controls the update rate of the pointing robot.
- **model type** --- This status field displays the type of algorithm the pointing robot actually uses. Possible data sources may be Keplerian elements (NASA 2-line format), Intelsat ephemeris data or list of positions defined in a text file.
- **model age** --- Displays the age of the data used by the robot. With ephemeris data sets this is the time since the epoch of the data set. For file tracking the time stamp of the last record in the table is used as the epoch.

Below these parameter fields a couple of fault indicators (red lamps) are located on the screen. They report any faults of the tracking engine. The meaning of the individual faults are:

- BEACON LEVEL --- The beacon level is below the threshold and one of the tracking modes STEP or ADAPTIVE is selected. In STEP track mode the antenna freezes in this situation, in ADAPTIVE mode the antenna is moved along the model which has been calculated before the beacon disappeared.
- PEAKING AZ/EL --- The step track peaking failed for one of the axes. The tracking engine reports this fault if the position / beacon level pairs measured during the tracking cycle

does not match the antenna pattern curve sufficiently. This may happen if the beacon values are too noisy or if there is no beacon reception at all and the threshold is adjusted too low.

- MODEL FIT --- If the variance of the measured peaks to the calculated model exceeds the 'jitter threshold' value, this fault is raised. This is an indication that either the measurement values are noisy or the model type is not appropriate for the satellite's motion.
- MODEL RESET --- If the ODM recognizes a 'MODEL FIT' fault at three consecutive tracking cycles, it presumes the model to be invalid and resets the model. This voids all recorded step track positions (they remain visible in the diagrams) and resets the tracking to the learning phase where no model is available.

3.14.2.4 The ACU Subsystem Menu

The icons below the tracking diagram are the ACU subsystem menu. They give access to some less frequently used parameters of the antenna control system.

Clicking to the ODM icon opens a window which permits access to all parameters of the ODM directly. Also the the color of the icon signals the fault state of the ODM device. With the ODM device window there are a couple of parameters which control the behavior of the tracking subsystem. Chapter 'The ODM Detailed Parameter Screens' gives a detailed description of them.

The BCRX icon opens a window directly to operate / configure the beacon receiver attached to the ACU. This window is required to set some beacon receiver parameters which are not accessible thru the application main screen.

The 'VIEW PLOT' button opens a separate graphical Y/T display window which gives more detailed information about what the tracking engine did during the recorded hours. It provides variable scales, and permits to display additional information. The display shows either the azimuth or the elevation pointing found at the beginning of each tracking step together with the beacon level measured at this time. The following additional information may be included to the diagram:

- --- This button has no function in this configuration.
- --- The main pointing graph (light green) always shows the pointing the antenna was really moved to. This is the smoothed position if smoothing is activated. If this toolbar button is pressed, the diagram includes the evaluated peak positions as a faded, dark green graph.

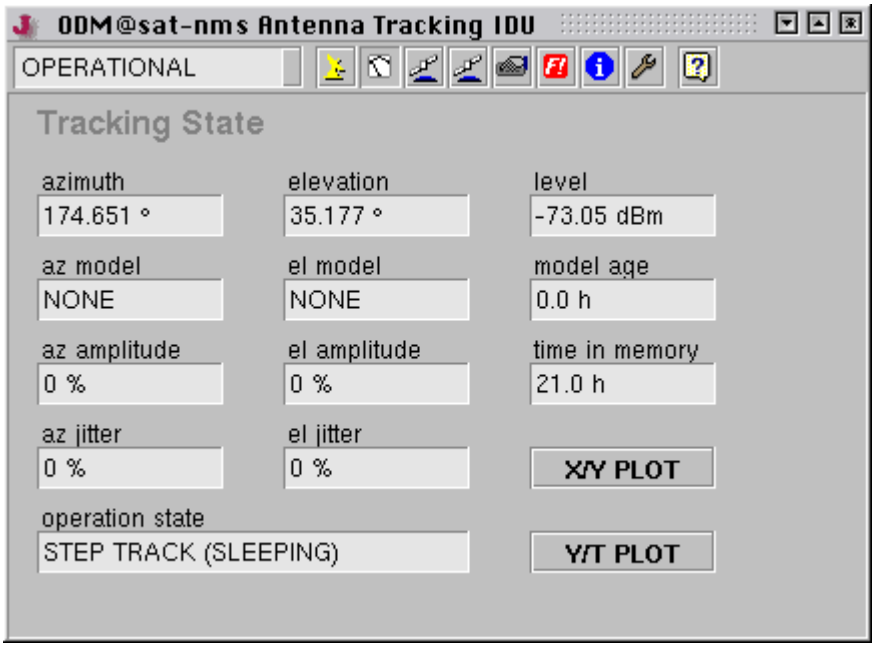
Clicking to the 'SATELLITE SELECT' button opens a window which lets you store and recall the position and the tracking parameters of several satellites. Also the ephemeris data for the pointing robot are entered here. Chapter 'The Satellite Select Screen' explains the functions of

this screen.

3.14.2.5 The ODM Detailed Parameter Screens

The ODM detailed parameter screens are part of the M&C device window for the SatService ACU/ODM. This window integrates all status and control parameters which are available for the ODM. To monitor and control the tracking functions of the ACU, the following two screens are important:

Clicking to the meter symbol in the toolbar shows the tracking state page. It displays a number of state parameters which report what the tracking engine actually does.

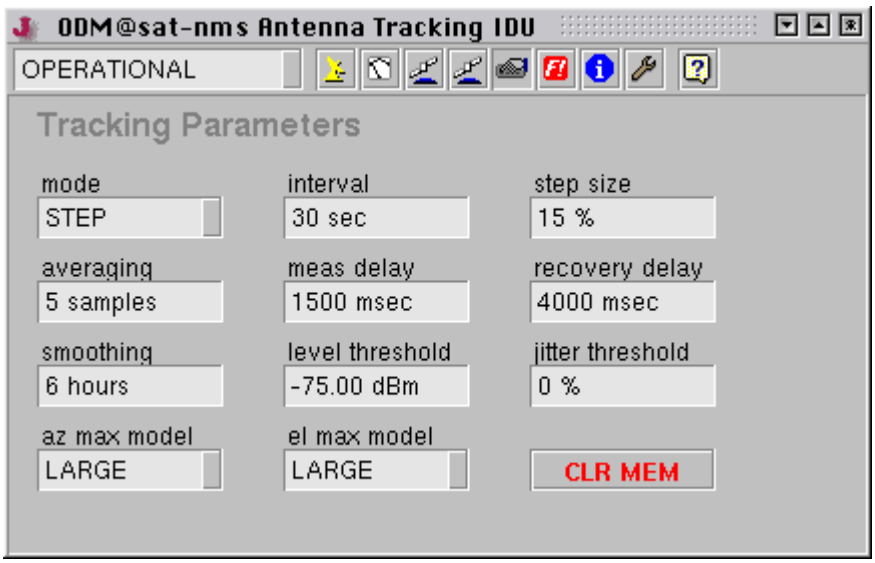


- **azimuth** --- The actual azimuth pointing.
- **elevation** --- The actual elevation pointing.
- **level** --- The actual beacon level (read from the ODM's analog input).
- **az model** --- The model type the ODM uses for the azimuth axis in ADAPTIVE mode if the beacon is lost. The value is displayed all the time, even if the beacon level is OK.
- **el model** --- The same for the elevation axis.
- **model age** --- This displays the time since the adaptive models have been updated the last time. The value normally is zero, if the beacon disappears it counts the time the antenna is tracked using the model.
- **az amplitude** --- The amplitude of the azimuth motion as a percentage of the antenna's 3 dB beamwidth. The ODM approximates the motion by a simple sine function to determine the amplitude.
- **el amplitude** --- The amplitude value for the elevation axis.
- **time in memory** --- The time span covered by the tracking data the ACU has in it's memory.
- **az jitter** --- The variance of all steptrack positions in memory with respect to the modeled values (azimuth). This is a measure how good the model fits. The jitter value is in percent if the antenna's 3 dB beamwidth. The 'jitter threshold' setting refers to the 'az/el jitter

values.

- **el jitter** --- The same for the elevation axis.
- **X/Y PLOT** --- Clicking to this button opens a separate window showing the same X/Y diagram as displayed in the main window.
- **operation state** --- Shows what the tracking engine actually is doing.
- **Y/T PLOT** --- Clicking to this button opens separate window showing the same X/T diagram as can be accessed from the main window.

Clicking to the device control symbol in the toolbar shows the tracking parameter page. Beside the tracking parameters shown in the application main window this page displays all adjustable parameters of the tracking subsystem.



- **mode** --- The tracking mode parameter selects the tracking method, the ACU actually uses. Possible selection are:
  - **OFF** --- No tracking is performed.
  - **STEP** --- Step track mode. In regular intervals, the antenna performs small search steps to optimize the pointing.
  - **ADAPTIVE** --- The adaptive tracking mode works the same way as step track, but it additionally is capable to predict the satellite's position when the beacon reception fails. It computes mathematical models of the satellites motion from the step track results recorded over a certain time.
  - **PROGRAM** --- The program tracking mode is offered here only for compatibility purposes only. In conjunction with the IDU the pointing robot should be used rather than the program track mode of the ODM.

A full description of the tracking functions used in the sat-nms ACU is given in chapter '5.0 Theory Of Operation' of the ACU/ODM manual.

- **interval** --- The interval time specifies how often the ACU shall perform a step track cycle. The value is to be entered in seconds. In fact, the parameter does not specify a cycle time but the sleep time between two tracking cycles. This means, the true cycle time is the time the ACU needs to perform one step track cycle plus the time entered

here. 300 seconds (5 minutes) is a good starting value for this parameter. Inclined orbit satellites probably will require a shorter cycle time, very stable satellites can be perfectly tracked with one step track cycle every 15 minutes (900 seconds).

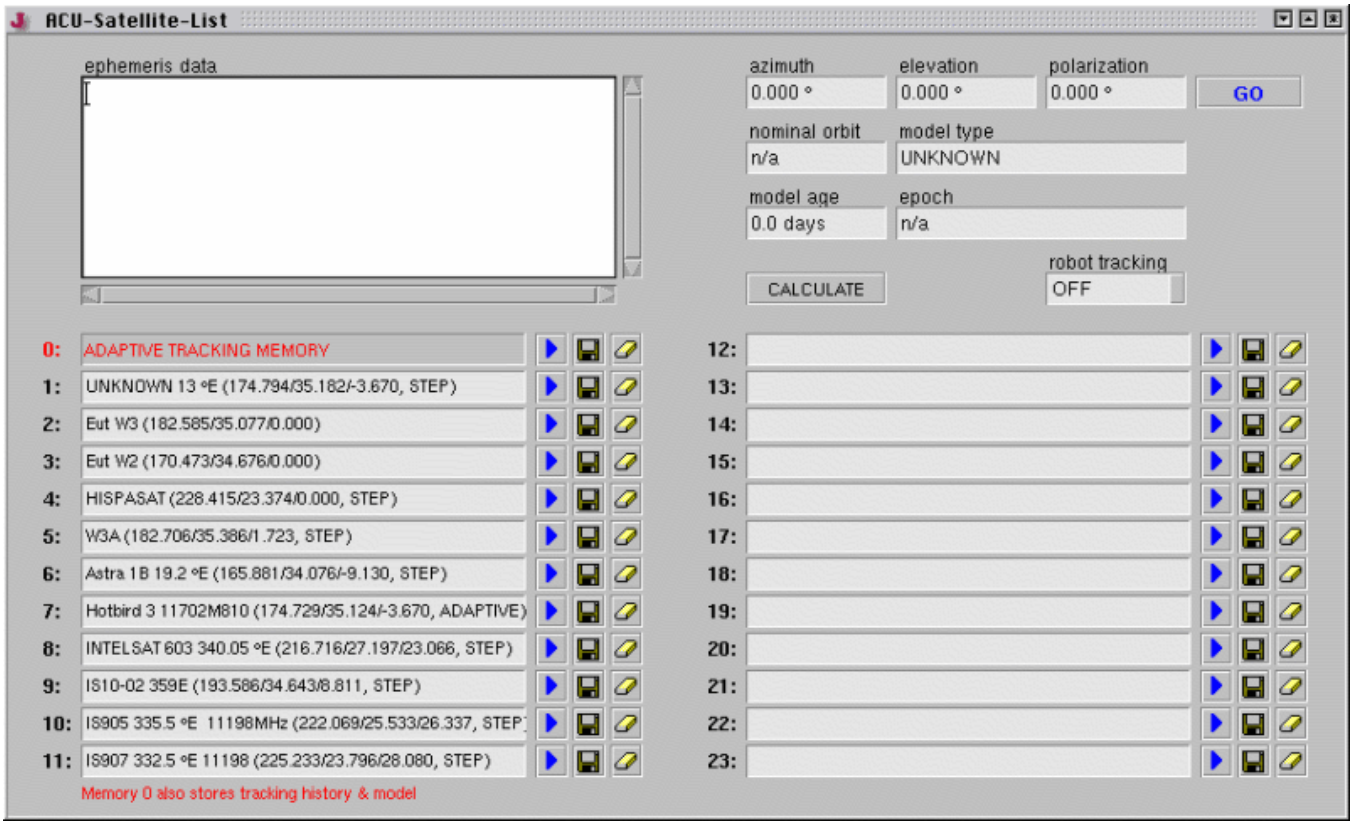
- **stepsize** --- The tracking step size is a very important parameter for the performance of the tracking. It defines the size of every de-pointing step, the ACU makes in order to find out where the optimal antenna pointing is. Setting too high values will cause significant signal degradations during the step track cycle because the antenna moves a too large amount away from the satellite. Setting the value too small will let the beacon level jitter mask the level differences caused by the test steps, the antenna will not track the satellite properly.The step size is specified as a percentage of the antenna's half 3dB beamwidth. The ACU calculates the beamwidth from the antenna diameter and the beacon frequency. Expressing the step size in this relative way keeps the value in the same range, regardless of th type of antenna. The recommended value for this parameter is 15-20%. You may want to start with 20% and try to reduce down to 15% if the signal degradation during tracking becomes too high.The tracking step size is a common parameter for both axes. If both axes behave differently, you can tweak the antenna diameter settings in the setup. Specifying a larger diameter makes the ACU using a smaller step size for this axis.If the tracking step seem to be completely out of range, you should check if the beacon frequency is set properly. The frequency must be the true receive frequency at the antenna, entered in MHz, not an L-band frequency or other IF.
- **averaging** --- When measuring the beacon level, the ACU takes a number of samples and averages them. The standard value of 5 samples normally should not be changed. Larger values will slow down the ACU execution cycle.
- **measdelay** --- During a steptrack cycle, the ACU positions the antenna to a certain offset and then measures the level. Between the moment when the antenna reached commanded position and the beacon level measurement the ACU waits some time to let the beacon level settle. The optimal delay value depends on the beacon receiver's averaging / post detector filter setting and is a quite critical for the steptrack performance.If the delay is too short, the beacon voltage does not reach its final value, the steptrack does not properly recognize if the signal gor better or worse after a test step. If the delay is too long, the impact of fluctuation to the measures level grows and may cover the small level difference caused by the test step. With the sat-nms LBRX beacon receiver, best results are achieved if the receiver is set to 0.5 Hz post detector filter bandwidth and a measurement delay of 1500 msecs
- **recoverydelay** --- After the the ACU has done the tracking steps for the elevation axis, it waits some time before it starts tracking the azimuth axis. This is to let the beacon level settle after the final position has been found. A typical value for this parameter is 4000 msecs
- **smoothing** --- This parameter controls the smoothing function. Setting it to zero disables smoothing. Smoothing lets the ACU point the antenna to positions evaluated from a simple model calculated from the step track peaks of the recent few hours. To understand the functionality of the smoothing function you may want to read chapter '5.3.3 Smoothing' in the ACU/ODM manual.
- **levelthreshold** --- If the beacon level falls below this threshold value, the ACU does not perform a step track cycle. If the level falls below the threshold during the steptrack cycle, the cycle gets aborted.If the ADAPTIVE tracking ist enabled and there is enough

data in the tracking memory, the ACU computes a mathematical model from the stored data and predicts the antenna pointing position from the extrapolation of the model. If the tracking mode is set to 'STEP', the ACU leaves the antenna where it is if the beacon level drops below the limit.Adjusting the threshold level that adaptive tracking is switched as expected must be done carefully and may require some iterations, specially if the beacon is received with a low C/N. A good starting value for the threshold is 10 dB below the nominal receive level or or 2 dB above the noise floor the beacon receiver sees with a de-pointed antenna, whatever value is higher.To turn off the monitoring of the beacon level (this in fact inhibits the adaptive tracking), simply set the threshold the a very low value (e.g. -99 dBm)

- **jitterthreshold** --- If the jitter value of at least one axis exceeds this threshold, the ACU raises an 'model fault'. If this happens three consecutive times, the ACU resets the models of both axes. Adaptive tracking will be possible not until 6 hours after this happens.A typical threshold value is 20%, this will detect very early that a model does not fit to describe the satellite's motion. If this value causes false alarms too often, you may want to raise the threshold to 50%. Setting it to 0 switches the threshold monitoring completely off.
- **azmaxmodel elmaxmodel** --- These settings let you limit the adaptive model to a simpler one, the ACU would choose by itself. The maximum model type can be set individually for each axis. Normally you will set both axes to 'LARGE', which leaves the model selection fully to the the ACU's internal selection algorithms.In cases where the ACU seems to be too 'optimistic' about the quality of the step track results, the maximum model on one or both axes may be limited to a more simple and more noise-resistant model. Specially inclined orbit satellites which are located close to the longitude of the antenna's geodetic location may require this limitation for the azimuth axis. With such a satellite, the elevation may move several degrees while the azimuth shows almost no motion.
- **CLRMEM** --- Clicking to this button clears the tracking memory. You should do this when you start to track a new satellite. Clearing the tracking memory about half an hour after tracking started significantly improves the quality of the first adaptive tracking model which will be evaluated after 6 hours of tracking. This is because the model does not get disturbed by the first search steps the antenna does until the optimal pointing to the satellite is found.

3.14.2.6 The Satellite Select Screen

The satellite select screen combines two functions for pointing the antenna to a satellite's position: The entry of ephemeris data to the pointing robot and the administration of the ACU's target memory locations.



The upper part of the screen is used to feed the pointing robot with the ephemeris data this requires to point the antenna to the satellite's position.

The usage is very simple: Copy the ephemeris data from e.g. your web browser window into the 'ephemeris data' field and click to the 'CALCULATE' button. The software tries to identify and parse the data and sets the informational fields on this right site to the values computed from the data. The 'azimuth' / 'elevation' / 'polarization' files show the antenna pointing which actually lets the antenna look to the satellite. The values below describe the type and the age of the ephemeris data set.

The pointing robot is capable interpret several types of ephemeris data including Keplerian elements in NASA 2-line format, Intelsat data, plain orbit positions and tables of az/el/pol value triples. The description of the Pointing-Robot logical devices describes the recognized data formats in detail.

The button 'GO' lets you make the pointing robot assist you in finding a satellite: Enter the satellite's nominal orbit position (or it's ephemeris data, if known), then click to 'CALCULATE', verify the calculated antenna pointing anf finally click the 'GO' to let the antenna move to this position.

The lower part of the screen manages the permanent storage of satellite parameters. Up to 24 satellites may be stored. The target memories store the antenna pointing and all tracking parameters. They are physically stored at the ODM's flash memory. If the ODM gets replaced, the stored positions are lost.

The ACU also stores the beacon receiver settings together with a target. These are stored at

the beacon receiver itself. The ODM, when receiving a command to store or recall a target memory, sends a command to beacon receiver to do the same.

- --- Recalls this preset. The antenna is moved to the stored position, all tracking parameter are set as stored and the beacon receiver's parameters are set from the values stored with the preset. The tracking memory is cleared if a target memory is recalled. Only if the target number 0 is recalled, the tracking memory is restored to the state when this target was saved.
- --- Stores the actual settings in this preset. The satellite name, the actual pointing and all tracking parameter are stored in the preset. The beacon receiver parameters are stored as well.
- --- The preset memory with this number is deleted.

The target memory location 0 is a special one. Unless the other memory locations, this first one stores the contents of the tracking memory and all temporary tracking data like models etc. together with the position and parameters.

This makes the first memory location dedicated to the short time storage of a satellites tracking state, when another satellite shall be tracked for a couple of hours. Example:

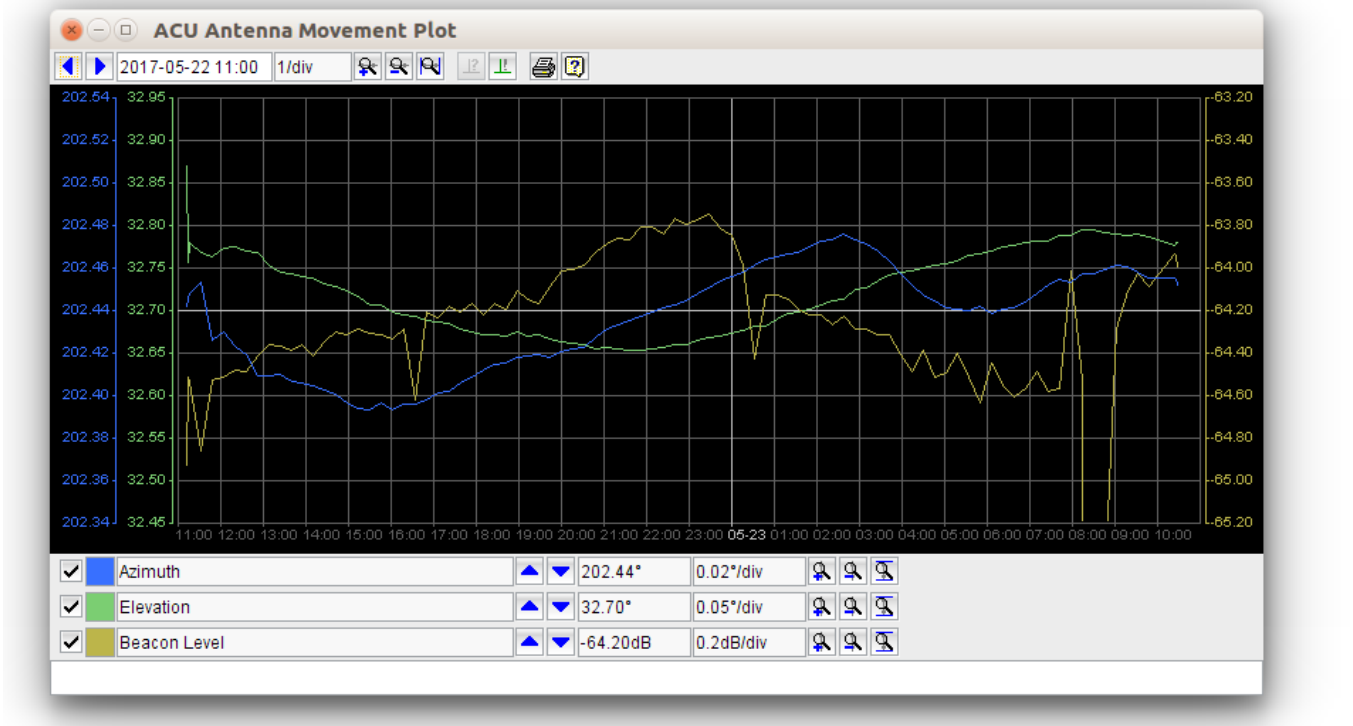
The antenna is tracking satellite A for a long time, the tracking memory is filled with 70 hours of step track data for this satellite. During a maintenance phase, the antenna shall temporarily track satellite B. With the target 0 memory you would handle this situation as follows:

1. Store the actual state as target 0 *before* the antenna is moved to point to satellite B.
2. Move the antenna to point to satellite B, either by recalling the target memory for this satellite or by pointing the antenna and clearing the tracking memory manually.
3. Let the antenna track satellite B for a couple of hours as needed.
4. No return to satellite A by recalling the target memory 0. This points the antenna back to satellite A and restores the tracking memory data for this satellite.

You should consider that it only makes sense to reuse the stored tracking data if there is less than twelve hours of discontinuity in tracking the satellite. If too much data is missing, clearing the tracking memory wil give the more reliable results.

3.14.2.7 The Y/T Tracking History Viewer

The sat-nms ACU IDU software provides a facility that permits to display the recorded tracking data in a versatile Y/T diagram viewer. This viewer may be launched by clicking to the "Y/T Plot" button either in the *ACU Main Screen* or in the device screen of the Satservice ACU-ODM.



The window shows a diagram of the recorded tracking data, the fields and buttons in the tool bar permit to modify the appearance of the diagram as needed. Clicking into the diagram shows a marker which displays a readout of the value at the selected screen position in the status bar at the bottom of the screen.

Top Toolbar

- --- Moves the plot two divisions to the left, lets you see older data.
- --- Moves the plot two divisions to the right, lets you see younger data.
- --- Shows/edits the start time. The displayed time corresponds to the leftmost grid line in the diagram.
- --- Zooms in the time scale, shows a smaller section of the data more detailed.
- --- Zooms out the time scale, shows a larger section of the data.
- --- Zooms out to show the recent 24 hours of recorded data.
- --- If pressed (it's a latching button), the search step beacon degradation is shown together with the beacon level. This function is not available with the SatService ACU-ODM.
- --- If pressed, the AZ/EL pointing determined by the peaking algorithm are shown as a second curve. The main curve shows the pointing angles where the antenna was moved to, including and optimization or smoothing applied.
- --- Prints the diagram as actually shown.

- --- Shows this manual page.

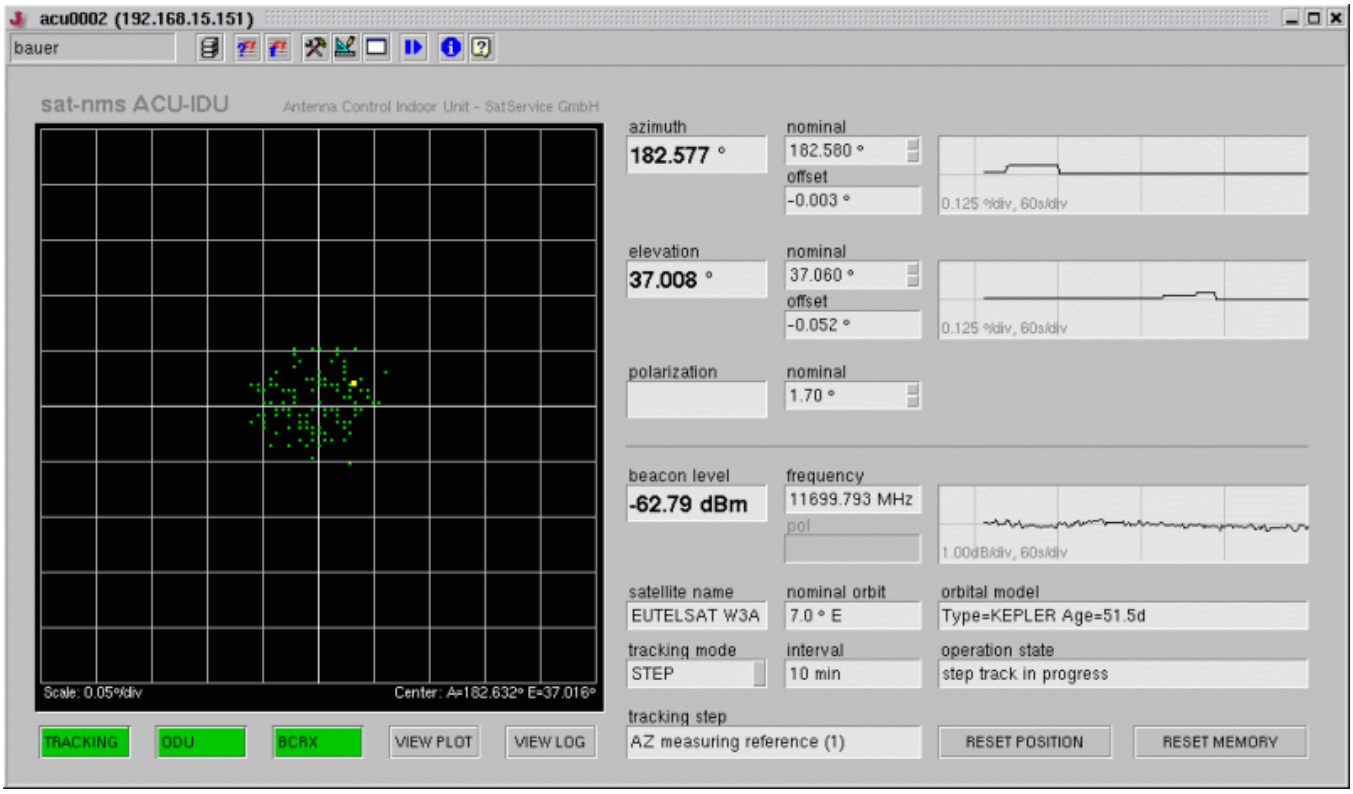
Bottom Toolbars

Below the diagram, the History File Viewer provides three toolbars to modify the scaling of the traces "Azimuth", "Elevation" and "Beacon Level" shown in the diagram.

- --- The first control in each toolbar selects if the trace shall be drawn.
- --- Shows the color if this trace, clicking to the color fields opens a dialog which lets you change the color of the trace.
- --- Shows the name / description of the trace.
- --- Moves the trace upwards one division.
- --- Moves the trace down one division
- --- Shows the y-scale reference level and the actual scaling. The y-reference level is the value corresponding to middle grid line.
- --- Zooms in the y-scale for this trace.
- --- Zooms out the y-scale for this trace.
- --- Zooms to show the min/max values (autoscale). This refers to the entire data, not only to the part actually shown.

3.14.3 The ACU Main Window (with 3rd party controller)

The main screen of the sat-nms ACU Indoor Unit shows the main parameters of the antenna tracking control similar to the front panel of a oscilloscope. The left part of the window contains a diagram which shows the antenna movements of the past 48 hours. The most important settings and status displays are located at the right.

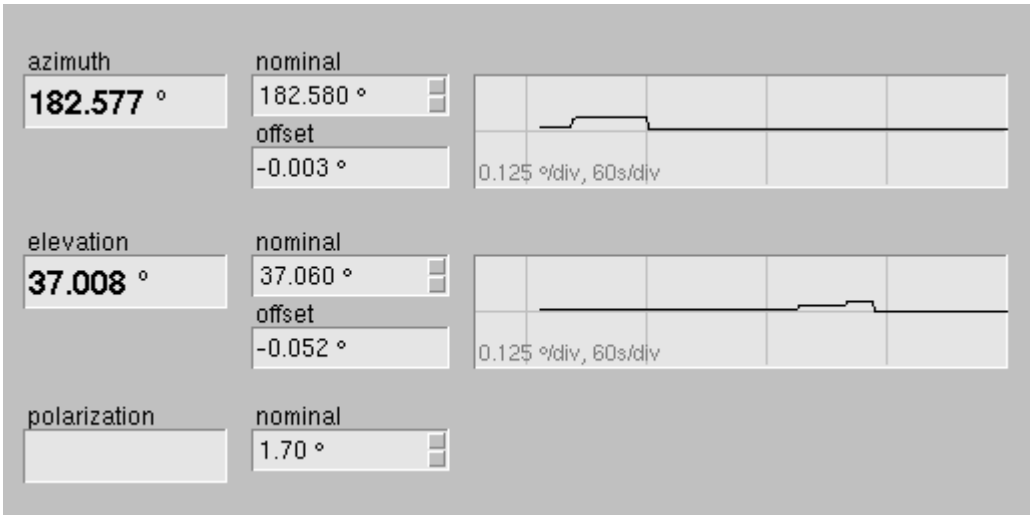


The toolbar on top of the window gives access to user login/logout, the event log and some other functions. As the ACU indoor unit inside principally is a M&C system, the toolbar is exactly the same as for the M&C Main Window .

At the top of the right part of the window thepointing display is located. It shows the actual antenna pointing and a strip chart of the recent history of antenna movements. Below the pointing display you find the beacon level display It reports the actual beacon level in a similar way. At the bottom of the right side the window contains an area with the tracking status display . Finally, below the tracking diagram on the left side, the ACU main screen contains the ACU subsystem menu . These five buttons give you access to subsystem screens with all detail parameters of the ACU / tracking.

3.14.3.1 The Pointing Display

The area in the upper right part of the window shows the actual antenna pointing and provides entry fields to move the antenna to a given position. The example below is taken from an antenna which is not equipped with a polarization motor, the polarization angle display is empty therefore.



The fields labeled 'azimuth', 'elevation' and 'polarization' show the actual antenna pointing as read out from the angle encoders. For azimuth and elevation there is also a strip chart display of the angle. If you choose a high resolution scale for the chart, you can watch the antenna doing the tracking steps in the chart. Click with the right mouse button into the chart to get a pop-up menu for scales and other display options.

The 'nominal' fields define the nominal antenna pointing for the actual satellite. The nominal pointing describes the center of the satellite's path over 24 hours.

- If you store the antenna pointing and the tracking parameters as a ACU preset, the software stores the nominal pointing. If you recall the preset, the software moves the antenna to the stored pointing and starts tracking there in the operation mode stored with the preset.
- The antenna moves to the nominal position, if you click to the 'RESET POSITION' button.
- The absolute tracking limits configured in the TRACKING subsystem refer to the nominal antenna pointing.

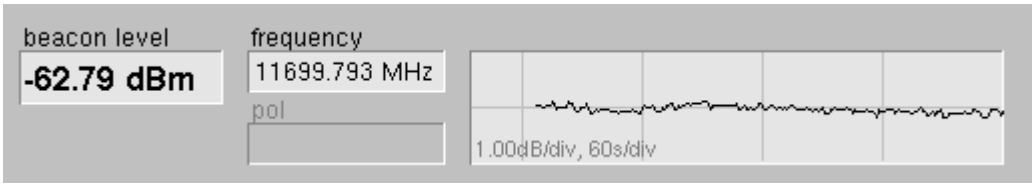
If you change a nominal pointing value while the antenna has tracking switched on, this only moves the reference point for the tracking limits. The antenna is under control of the tracking engine and does not move due to this change.

If the ACU tracking mode is set OFF, the antenna behaves differently. In this case changes of the nominal value let the antenna move immediately to the selected position.

Finally, the 'offset' fields indicate the actual difference between the real antenna pointing and the nominal position.

3.14.3.2 The Beacon Level Display

Below the antenna pointing, the ACU shows the actual beacon level. There is also a strip chart display of the level. The chart moves with the same speed as the angle charts above, so you directly can watch the influence of tracking steps to the beacon level. A click with the right mouse button into the chart shows a pop-up menu for scales and other display options.

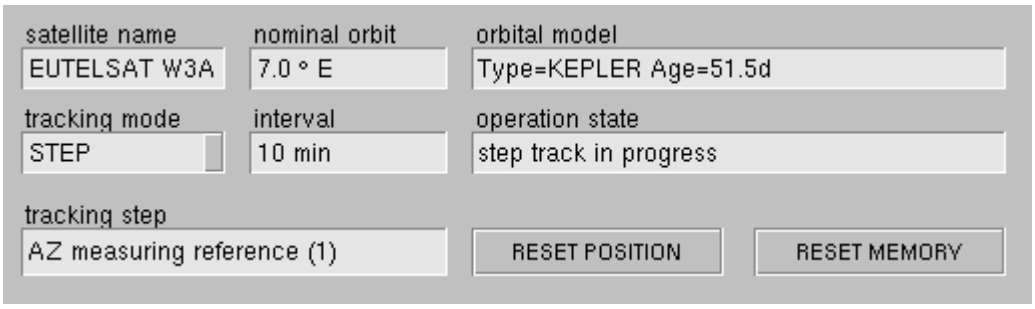


The screen also provides fields to set the beacon frequency and the receive polarization (the latter is not supported by antenna where the example picture has been taken from). Changing the values in these fields changes the beacon receiver setting immediately.

If you are using a third party beacon receiver which is not controlled by the ACU indoor unit, you nevertheless have to set the beacon frequency here. This is because the tracking software takes the frequency into account when estimating the antenna receive pattern and from this the tracing step sizes.

3.14.3.3 The Tracking Status Display

The lower right area of the ACU main window shows the tracking status display and the basic tracking mode parameters.



The table below gives a short description of each parameter in this area. The all refer to the TRACKING subsystem , this chapter gives a more detailed description of the tracking.

- **satellite name** --- The name of the satellite you are tracking.
- **nominal orbit** --- The nominal orbit position of the satellite.
- **orbital model** --- The type and age of the orbital model the ACU uses for the PROGRAM track mode. During ADAPTIVE tracking, the the number of hours in memory are displayed here.
- **tracking mode** --- With this parameter you set the basic tracking mode. Available modes are OFF, STEP, ADAPTIVE and PROGRAM..te .ts **interval**With this parameter you define in which intervals the ACU shall perform tracking steps.
- **operation state** --- This field shows the general operation state of the tracking engine.
- **tracking step** --- This field tells what the tracking engine is doing actually in this moment.
- **RESET POSITION** --- Moves the antenna back to the nominal pointing values. If you do this in ADAPTIVE or in PROGRAM track mode, the antenna will return to the estimated pointing with the next tracking step. With STEP track mode, the antenna starts tracking at the nominal pointing if you click to this button
- **RESET MEMORY** --- Clicking to this button clears the tracking memory, the ACU

remembers to estimate the antenna pointing in ADAPTIVE mode. You should do this if you change tracking to another satellite or if you recognize a significant repositioning of the satellite.

3.14.3.4 The ACU Subsystem Menu

The five icons below the tracking diagram are the ACU subsystem menu. Although looking unimposing, this menu is very important as it gives you access to all parameters of the ACU's subsystems.

Clicking to the TRACKING icon launches a window which shows all parameters of the tracking subsystem. The color of the icon tells about the fault state of the tracking engine (green = OK, red = FAULT). You open the TRACKING subsystem window in order to

- save and recall satellite presets.
- adapt low level tracking parameters like step sizes, averaging times etc.
- enter ephemeris data for antenna positioning or PROGRAM track mode.

The ODU icon opens a window to operate / configure the antenna positioner device. Normally it will not be necessary to access the positioner directly in this ACU configuration, moreover, it might mess up the tracking process if you command the antenna directly. There are however a few situations where you want to see all antenna parameters:

- The positioner shows a fault (red icon) and you want to know what the reason for the fault is.
- You want to move the antenna manually in small steps, e.g for the pointing calibration or for measurements. The SatService ACU ODM device window provides step movement buttons which help you to do this efficiently.

The BCRX icon opens a window directly to operate / configure the beacon receiver attached to the ACU. This window is required to set other beacon receiver parameter than the frequency or receive polarization.

The 'VIEW PLOT' button opens a separate graphical Y/T display window which gives more detailed information about what the tracking engine did during the recorded 48 hours. It provides variable scales, and permits to display additional information. The display shows either the azimuth or the elevation pointing found at the beginning of each tracking step together with the beacon level measured at this time. The following additional information may be included to the diagram:

- --- If this toolbar button is pressed, the diagram shows the search step and the beacon level change resulting from this as a dimmed peak at the left edge of each data point. You can easily monitor the signal degradation which is caused temporarily by the tracking search steps with this function enabled.
- --- If this toolbar button is pressed, the diagram includes the evaluated peak positions.

In STEP track mode, this actually is the location where the antenna gets moved to. In ADAPTIVE tracking mode, the antenna moves to the location resulting from the mathematical model which is computed from the step track peaks. In the latter case, the peak positions may be interpreted as the actual difference between the modeled pointing and the peak position.

Clicking to the 'VIEW LOG' button opens a window where the program shows a live report what the tracking engine actually is doing. The log in this window is cleared with each new tracking step. If 'Y/T PLOT' diagram shows many unsuccessful tracking steps, watching the log reports may give an explanation why the tracking engine discards so many steps.

3.15 Specialized Client Applications

In addition to the standard Java client, the sat-nms software provides a number of client applications specialized to certain tasks. This chapter describes the operation of these (legacy) appliacations. Thy are:

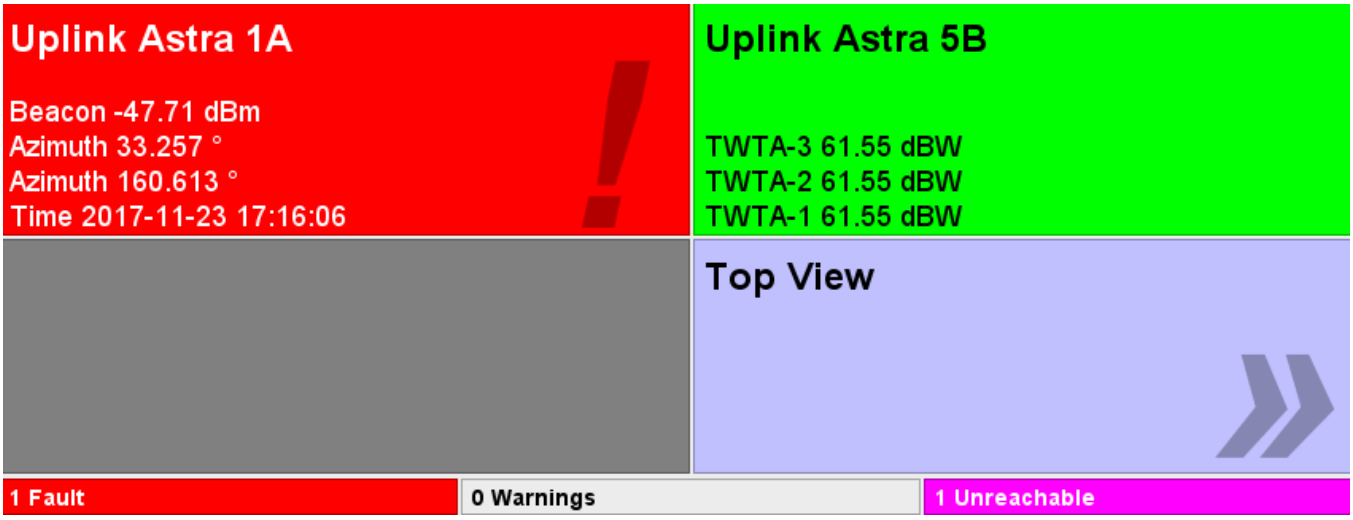
- The MasterClient Application The sat-nms MasterClient-2 is a client program for multiple sat-nms M&C servers. It can be used in parallel to the standard M&C client program in order to organise certain M&C system under a common main GUI.
- The ServiceClient Application The sat-nms Service-Client is a client program for the sat-nms M&C server. It can be used in parallel to the standard M&C client program in order to organise certain devices and device parameters in "services".
- The ViOIPClient Application The sat-nms VioIP (video over IP) Software is an extension to the sat-nms M&C software which permits to manage video over IP transmissions in a given IP network. The software manages resources like the bandwidth of connections between two nodes of the network or the usage of endpoint devices.
- The VioIP Network Configuration Application This program permits to add or delete locations, endpoint devices or data lines between the locations.

3.15.1 Master Client 2

The sat-nms MasterClient-2 is a client program for multiple sat-nms M&C servers. It can be used in parallel to the standard M&C client program in order to organise certain M&C system under a common main GUI.

You can invoke the MasterClient 2 with the following instructions (e.g. in a shortcut):

javaw -cp client.jar satnms3.mc2.MasterClient [config-file-name].xml



3.15.1.1 MasterClient 2 XML Configuration File

The version 2 MasterClient program reads it's configuration an XML file. Structure and the meaning of the XML elements are described in the following chapters.

When the MasteClient program is started without a command line argument, it reads a file **MasterClient.xml** in the actual working directory to read its configuration Alternatively a file or path name may be given as the only command line argument, the program then will read this file instead.

XML Element Hierarchy

The diagram below shows the overall tree structure of the XML elements making up the MasterClient's configuration. Most elements may appear multiple times except the <masterclient> element.

```
<masterclient>         The root element
|
+-<global>            Definition of a global parameter
|
+-<peerlist>          List of all peers to M&C systems
| |
| +-<peer>            Definition of one peer to a M&C system
|   |
|   +-<parameter>       Definition of an optional parameter to be
|                   watched at the M&C
|
+-<grouplist>          List of all groups
| |
| +-<group>           Definition of one group
|   |
|   +-<member>         Definition of one member in the group
|
+-<screenlist>         List of all screens
  |
  +-<screen>           Definition of one screen
    |
    +-<row>           Definition of one row of buttons in this screen
      |
      +-<button>       Definition of one button
```

3.15.1.2 XML Element Description

The following paragraphs describe each element which may occur in the configuration file with its possible children and attributes.

- <masterclient> --- The **<masterclient>** element is the root element of the XML tree. It contains all elements which make up the the master client's configuration. Elements outside the **<masterclient>** element are ignored.
- <global> --- The **<masterclient>** element may contain zero or more **<global>** elements. These elements define global settings of the master client program. They don't have child elements but they each have two attributes which both are mandatory: *key* and *value* .The following keys are defined:
  - fontSize --- The size (pixels) of the font used for the button labels.
  - fontSizeParam --- The size (pixels) of the font used for parameters in M&C buttons.
  - title --- The text to shown in the window title bar
  - okColor --- The color of a button standing for a M&C or group which is in state *OK* .
  - faultColor --- The color of a button standing for a M&C or group which is in state *FAULT* .
  - warningColor --- The color of a button standing for a M&C or group which is in state *WARNING* .
  - linkColor --- The color of a button which simply changes to another screen without

showing a fault state
- unreachableColor --- The color of a button standing for a M&C or group which is in state *UNREACHABLE* .
- disabledColor --- The color of a button standing for a M&C or group which is actually disabled.
- autoTextColor --- One of *true* or *false* . With *true* the button text is drawn black bright background colors, white for dark background colors for better contrast. With *false* the text is always drawn black.
- showSymbols --- One of *true* or *false* . If set *true* , the state of a M&C or group button is not only symbolized by its color but also by an exclamation mark for *FAULT/WARNING* or a question mark for *UNREACHABLE* . These symbols are drawn large in a faded color as part of the button background.
- priorityScheme --- One of *FAULT* or *UNREACHABLE* . Defines the priority of states a group uses to combine the states of its children. When set to *FAULT* , the priority is FAULT (highest) - WARNING - UNREACHABLE - OK (lowest).When set to *UNREACHABLE* the priority is UNREACHABLE (highest) - FAULT - WARNING - OK (lowest).A group always shows the highest priorized state one of its children shows.

Examples and default values for all keys:

```
<global key="fontSize" value="14" />
<global key="title" value="sat-nms Master Client 2" />
<global key="okColor" value="#00FF00" />
<global key="faultColor" value="#FF0000" />
<global key="warningColor" value="#FFFF00" />
<global key="linkColor" value="#8080FF" />
<global key="unreachableColor" value="#808080" />
<global key="disabledColor" value="#808080" />
<global key="autoTextColor" value="true" />
<global key="showSymbols" value="true" />
<global key="priorityScheme" value="FAULT" />
```

- <peerlist> --- The <peerlist> element contains the definition of all peers, each connecting to one M&C system and watching its state.
- <peer> --- The **<peer>** element defines one peer connecting to a M&C system and watching its state. The **<peer>** element may contain zero or more **<parameter>** elements which define parameters to watch additionally to the M&C summary fault state.Each **<peer>** element has a number of attributes which identify the element and define how to connect to the M&C system:
  - id --- A unique name identifying this peer. Groups and screen buttons use this identifier to refer to this peer / M&C. This attribute is mandatory.
  - ip --- The IP address of the M&C in dotted quad notation. This attribute is mandatory.
  - mode --- One of *enabled* or *disabled* This defines the the initial mode of this peer after program start. Enables peers operate normally and reflect their fault state as reported by the M&C. Disabled peers show a state *disabled* regardless of the fault state of the M&C. The mode attribute is optional, the program assumes *enabled* if

the attribute is missing.
    - clientpars --- Optional parameters added to the command line which launches the client program when the operator clicks to a button referring to this M&C. Typical contents for the clientpars attribute are user/password settings.

Examples:

```
<peer id="MNC1" ip="192.168.0.100" clientpars="-u satnms -p satnms -s default -q" />
<peer id="MNC2" ip="192.168.0.101" clientpars="-u satnms -p satnms -s default -q">
   <parameter id="EIRP-2.on" label="TX" />
</peer>
```

- <parameter> --- The **<parameter>** element defines one parameter the peer shall watch at its M&C additionally to the summary fault. A **<peer>** element may contain an arbitrary number of **<parameter>** elements as children.A screen button referring to the peer shows the actual values of the defined parameters, starting with the first parameter near the bottom of the button rectangle, the second one above this and so on.Parameters are shown with a fixed font size, depending on the button size they may be covered by the button label or completely ourside the button area, invisible for this reason.A **<parameter>** element contains these two attributes:
    - id --- The message ID of the parameter to watch.
    - label --- A label drawn on the screen button left of the parameter value

Example:

```
<parameter id="SYSTEM.time" label="Time" />
```

- <grouplist> --- The **<grouplist>** element contains the definition of all groups.
- <group> --- The **<group>** element defines one group within the **<grouplist>** . The group is named by a identifier, the group members are defined by **<member>** child elements within the **<group>** . The **<group>** element knows one (mandatory) attribute:
    - id --- A unique name identifying this group. Other groups and screen buttons use this identifier to refer to this group. Identifiers must be unique across **<peers>** and **<groups>** because both may appear as group members and state sources for screen buttons in the same way.

Example

```
<group id="GRP1">
  <member id="MNC1" />
 <member id="MNC2" />
</group>
```

- <member> --- The **<member>** element defines one group member within a **<group>** definition. The **<member>** element has one mandatory attribute selecting the **<peer>** or **<group>** as the group member:
    - id --- The existing name / identifier of the peer or group which shall be a member of

this group.

Example:

```
<member id="MNC1" />
```

When defining group members, care should be taken not to create circular references between groups: if e.g. a group defines itself als member or group A has group B as member and at the same time group B defines group A as member, this creates a race condition in the software, the master client won't work.

- <screenlist> --- The **<screenlist>** element contains the definition of all screens. The **<screenlist>** element may contain one attribute:
  - default --- The ID of the screen which shall be loaded when the program

Example:

```
<screenlist default="HOME">  ...  </screenlist>
```

- <screen> --- The **<screen>** element defines one screen of the MasterClient program. The MasterClient always shows one grid of N x M buttons. A screen is one set of such buttons shown at a time. Buttons referring to groups or simple link buttons may be clicked to switch to another screen, another set of buttons.A screen is defines as a sequence of **<row>** elements each containing a sequence or **<button>** elements. Buttons appear from left to right in a row, rows appear from top to bottom in the screen. The program scans the screen definitions at startup and evaluates the maximum number of rows and columns defined. The MasterClient always shows a grid of this size, filling it up with empty buttons if a particular screen or row is smaller than the evaluated size.The **<screen>** element contain one mandatory attribute:
  - id --- A unique name identifying this screen. The default attribute of the **<screenlist>** element and the **<target>** attribute of the **<button>** element refer to this identifiers to select this screen.

Example:

```
<screen id="HOME">  ...  </screen>
```

- <row> --- The **<row>** element defines one row of buttons in the screen. It has no attributes, contains zero or more **<button>** elements.
- <button> --- The **<button>** element defines one button within the row / screen. It contains up to three attributes:
  - src
  - label
  - target

Depending on the presence of attributes a button behaves differently

- A button without any attributes behaves as a dummy button or spacer

- A button containing a *target* attribute but no *src* attribute behaves as a link button, it simply switches to the target screen when clicked
- A button containing a *src* attribute referring to a **<peer>** element is a M&C button. It shows the fault state of the M&C and launches the client program for this M&C when clicked. It should not contain a *target* attribute.
- A button containing a *src* attribute referring to a **<group>** element is a group button. It shows the fault state combined by this group It must contain a *target* attribute which denotes the screen to be shown when the button is clicked.

3.15.1.3 Example showing a complete screenlist

```
<screenlist default="HOME">
 <screen id="HOME">
  <row>
   <button src="GRP1" label="Group 1" target="G1SCREEN" />
   <button src="GRP2" label="Group 2" target="G2SCREEN" />
  </row>
 </screen>
 <screen id="G1SCREEN">
  <row>
   <button src="MNC1" label="Station A" />
   <button src="MNC2" label="Station B" />
  </row>
  <row>
   <button />
   <button target="HOME" label="Top View" />
  </row>
 </screen>
 <screen id="G2SCREEN">
  <row>
   <button src="MNC3" label="Station C" />
   <button src="MNC4" label="Station D" />
  </row>
  <row>
   <button />
   <button target="HOME" label="Top View"/>
  </row>
 </screen>
</screenlist>
```

3.15.2 Service Client Application

The sat-nms Service-Client is a client program for the sat-nms M&C server. It can be used in parallel to the standard M&C client program in order to organise certain devices and device parameters in "services".

There may be multiple instances of the Service-Client running at the the same time. Operating

the software is fully multi-user capable, but adding/removing services should be done only from one computer at a time.

Service Definition A service is a combination of an arbitrary number of properties which each may be a up- or down converter, a splitter output, a combiner input, a fibre output or a test port. For each type of property there is a number of parameters which are monitored and - if applicable - controlled.

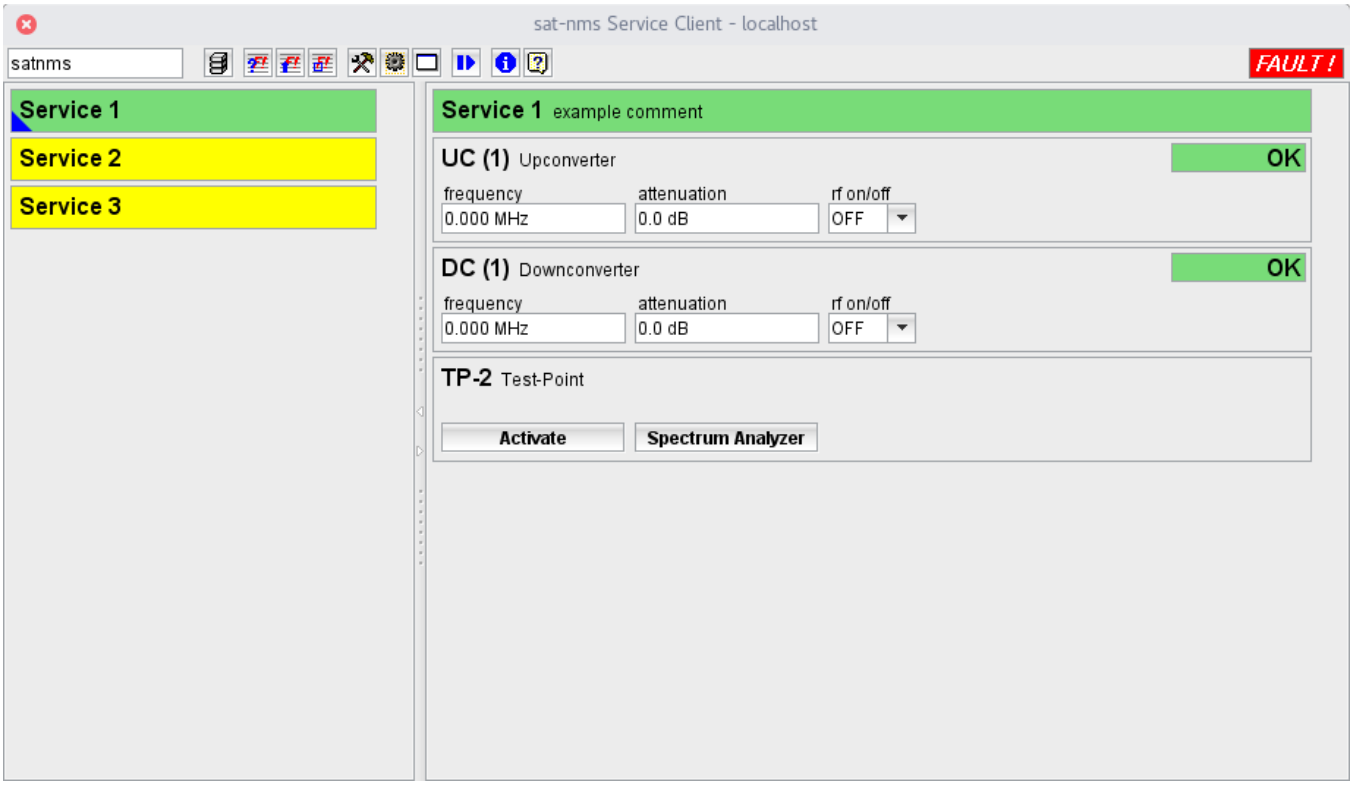
- property type --- monitoring only parameters, m&c parameters
- up-/downconverter --- fault flagfrequency attenuation signal on/off
- splitter output --- fault flag measured level (may be the common level at the splitter input)port on/off
- combiner input --- fault flag measured levelport on/off
- fibre output --- fault flag
- test output --- button/command to set all switch positions required to route the service output signal to this test point

There is a summary fault state for each service which is the "worst condition" of the properties contained in this service. This service summary fault states does not appear in the server's event log as it is evaluated in the client program and known only there.

Operation

The Service-Client window mainly consists of a list of available services at the left and the actual settings / readings of the selected service at the right. By clicking to a service name the service gets selected and its parameters are shown at the right. Both, service list and parameter panel automatically add vertical scrollbars if required.

The program inherits a lot of functions from the sat-nms standard M&C client. It uses the same login scheme as the standard M&C GUI, toolbar buttons for event report query, live event log and list of active faults work in the same way here.



At the left side the window shows a list of all defined services. The background color shows the actual fault state of the particular service (one of DISABLED/OK/WARNING/FAULT). The colors used for this are configurable. Clicking with the right mouse button on a service name opens a context menu at the mouse pointer containing some convenience functions like enabling/disabling the service summary fault or opening the configuration editor for this service. The service fault status is updated in near real time for all defined service which are not "disabled".

The right side of the window shows the parameters and state of the actually selected service. For each property belonging to the service, a field containing the property's name, parameters and fault state. The layout of these fields is defined by a XML configuration file supplied by SarService. The displayed name of a property consists of a label for the property type, the sat-nms device name referenced and - if applicable - an index identifying the unit within the device used for this service.

Status values and control parameters of the properties of the selected service are updated in near real time like with the standard M&C GUI. Control parameters can be changed in place without the need to open another window.

Tool Bar

The Service Client provides a toolbar at the top of the window which gives access to a number of global functions in the software. Below there is a short description for each of the toolbar buttons.

- --- The left field at the window's tool-bar shows the login-name of the user who is

actually logged in. You may click with the mouse to this field, the program pops up the login/logout dialog.

- --- Shows a drop down list with all devices installed in the M&C system. Selecting a device from this list opens it's Device window .
- --- Opens the Event Report window .
- --- Opens the Live Event Log window .
- --- Opens a list of currently active faults.
- --- Opens the Device Setup window to configure the device setup of the M&C system.
- --- Switches the service client window to "edit" mode, lets you add or delete services or change service definitions (see "Configuration" below)
- --- Opens a terminal window which connects to the debug / test port of the M&C server.
- --- Opens the Macro Management window which controls recording and replay of equipment setting macros.
- --- Shows the program version info.
- --- Shows this help page.

Configuration

Configuring the Service-Client means to add or remove equipment (properties) from an existing service definition or to add or remove complete service definitions. For this purpose the Service-Client provides a configuration mode which can be switched on from the main window's tool bar. The configuration mode permits to:

- Add a new service
- Delete an existing service
- Change the order in which the services appear in the GUI
- Change the name of an existing service
- Add a property to an existing service
- Remove a property from an existing service
- Change the order in which the properties appear in the GUI

Changes made in configuration mode are reflected immediately, e.g. an added property shows up as soon as it is added. There is no "save" button in the configuration mode, all changes which are made are permanent.

When the configuration of services is changed, all other opened service clients are informed to re-read the configuration from the server to get updated. This happens within 15 seconds after a configuration change has been made.

When adding a new property to a service, the software knows which sat-nms devices or units within devices are suitable for the selected purpose. The software also knows which properties may be used only once and cannot configured a second time.

The operator gets a list of actually possible devices which may be configured as the new property for the service and selects the hardware representation of the property from this list.

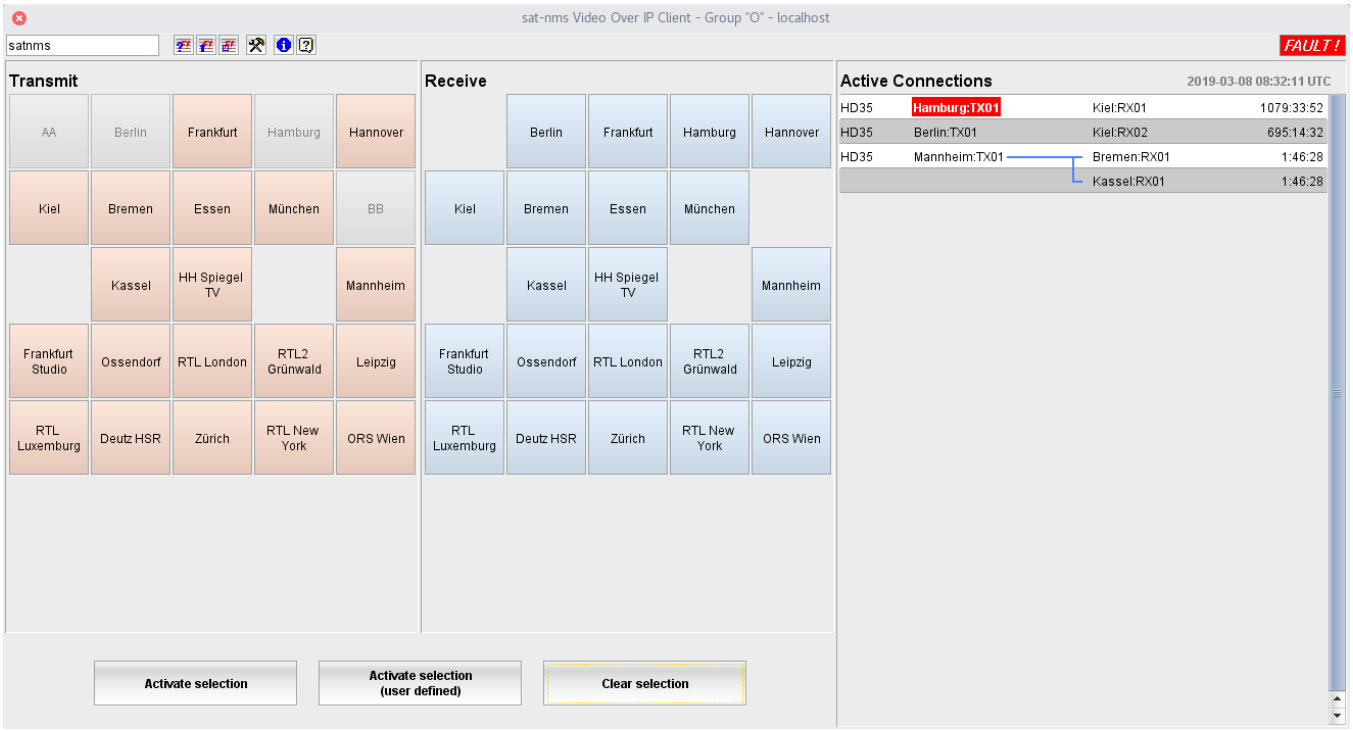
The knowledge, which types of devices are suitable e.g. as a combiner input, how many identical units of this a sat-nms device manages and through which parameter identifiers the

desired combiner port is addressed is defined in an XML configuration file which is supplied by SatService. By editing this file, more / new device types may be added to the list of known device drivers for the software.

3.15.3 VioIP Client Application

The sat-nms VioIP (video over IP) Software is an extension to the sat-nms M&C software which permits to manage video over IP transmissions in a given IP network. The software manages resources like the bandwidth of connections between two nodes of the network or the usage of endpoint devices.

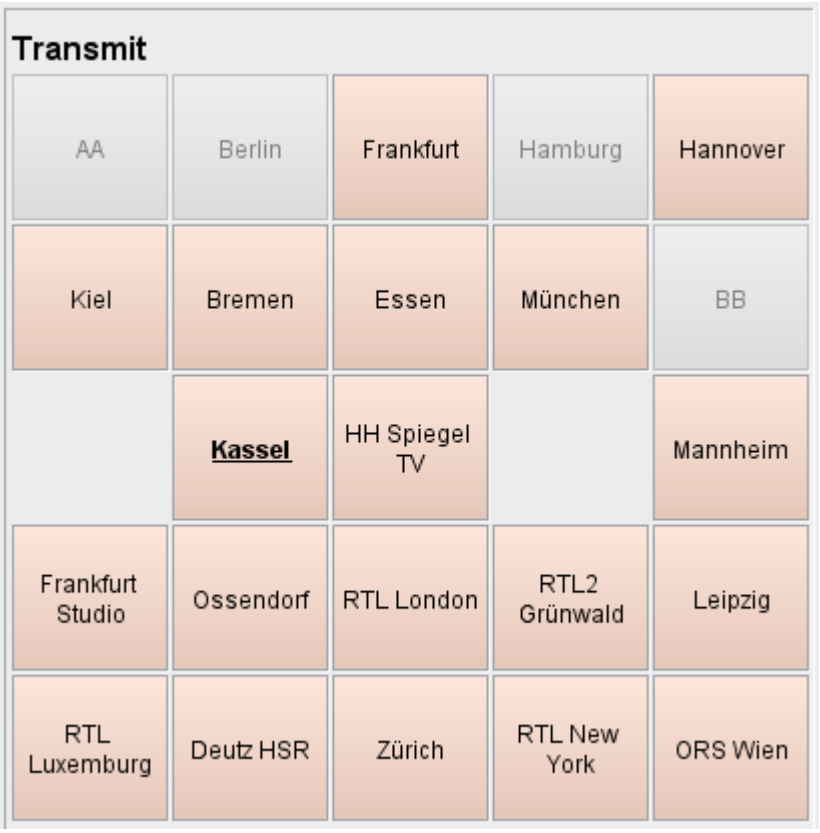
The sat-nms VioIP Software consists of an extension to the sat-nms M&C server with the resource database, the VioIP Client Application which is used to start or stop transmissions and finally the VioIP Network Configuration Application which permits to add or delete locations, endpoint devices or data lines between the locations. The following paragraphs describe the VioIP Client Application.



The VioIP Client Application window consists of three major parts: The TX location button array, the RX location button array and the list of active connections. At the top of the window there is a tool bar giving access to some global functions of the sat-nms software.

TX location button array

The array of TX locations shows one button for each location defined in the software which has at least one TX end point device configured. The button array is used to select the source of a connection to be started.



If there are no devices available to start a new connection from a particular location (because all devices are either in use, temporarily disabled or in communication fault state), the button is grayed out and cannot be pressed.

To select a location simply click the button. The name of the selected location prints bold/underlined to signal that this location is selected as the source for the connection to be started next. Only one TX location may be selected at a time, to change the selected TX location simply click another one. The selection mark then moves to this location. To unselect a location, click the selected location again or click the "Cancel selection" button below the button arrays.

RX location button array

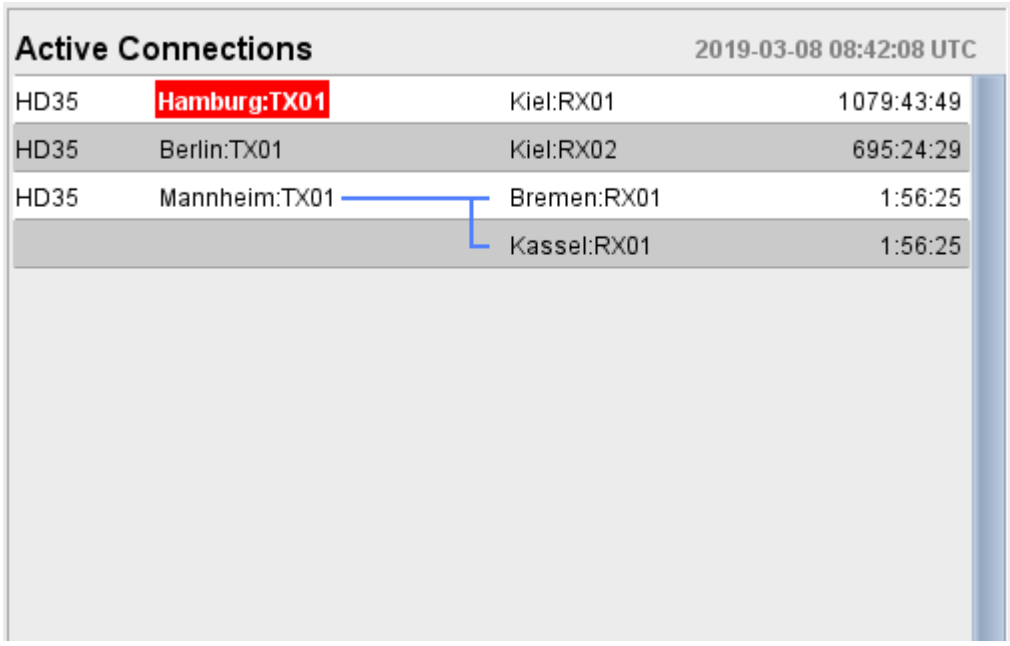
The array of RX locations shows one button for each location defined in the software which has at least one RX end point device configured. The button array is used to select the destination(s) of a connection to be started.

If there are no devices available to start a new connection to a particular location (because all devices are either in use, temporarily disabled or in communication fault state), the button is grayed out and cannot be pressed.

Using the RX location buttons is much like for the TX side, however the RX button array lets you multiple destination. Selecting more than one destination for an upcoming connection switches the operating mode to "multicast" which will send the program from the source location to all selected destinations.

Active connection list

The active connection list shows all active connection as a scrollable list, one connection per line. For each connection the video profile, the source (location and device) and the destination of the connection are shown. The last column shows the time how long this connection already is active.



Multicast groups are symbolized with blue lines, connecting one source to one or more destinations. If an end point device shows a fault while the connection is active, the name of the end point is drawn as white text on red background. Please note, that not all faults of the device are signalled in this way. In the logical VioIP-End-Point Device the faults are defined which are to be monitored while the connection is running.

Beside as a status display, the Active Connections list is used to protect, to un-protect and to stop running connections. Clicking with the right mouse button on a connection in the list shows a context menu which gives you several options, depending on the actual state of the connection. Below on this page you find paragraphs which describe elaborately how to terminate, protect or un-protect connections.

Tool bar buttons

The VioIP Client provides a toolbar at the top of the window which gives access to a number of global functions in the software. Below there is a short description for each of the toolbar buttons.

- --- The left field at the window's tool-bar shows the login-name of the user who is actually logged in. You may click with the mouse to this field, the program pops up the login/logout dialog.
- --- Opens the Event Report window .
- --- Opens the Live Event Log window .
- --- Opens a list of currently active faults.
- --- Opens the VioIP Network Configuration Application . This application lets you add or remove locations, end points, data lines an more. You must be logged as a user with privilege level 150 to use this function.
- --- Shows the program version info.
- --- Shows this help page.

Starting a connection

Starting a connection is straight forward, using the "Location Buttons" of the GUI. Click to a TX location and then to a RX location, finally click the "Activate selection" button below and the software starts a unicast connection between these locations. It uses the first available end point device at either location and the default video profile (bandwidth) for this.

You may select multiple RX locations by clicking the buttons in the RX location button array. In this case the software automatically uses multicast to transmit the program from the TX location simultaneously to all selected destinations.

If you accidentally selected an RX location, click it again to deselect it. This must be done before you activate the selection.

Terminating a connection

To terminate a connection click to the connection in the list with the right mouse button and select "Stop this connection" from the context menu popping up. With multicast connections there is also an option "Stop all connections of this multicast group".

Stopping a unicast connection means to turn off the transmission, switching all devices to stand-by and finally freeing the resources occupied by this connection (endpoint devices and used bandwidth). The same applies when stopping all connections of a multicast group or when stopping the very last connection of a multicast group.

Stopping a single connection from a multicast group means - unless this is the very last connection of this group which is still running - to switch of the receiver, freeing this as a resource and freeing the bandwidth which was used to send the signal to particularly this receiver.

Protecting a connection

Running connections may be protected from accidentally being terminated. Protected connections cannot be terminated directly, the protection must be removed before the connection can be terminated. Protected connections appear with a yellow colored background in the connection list.

To protect a connection, click to the connection in the list with the right mouse button and select "Protect this connection" from the context menu popping up. With multicast connections there is also an option "Protect all connections of this multicast group"

To remove the protection from a connection, click to the protected connection in the list with the right mouse button and select "Unprotect this connection" from the context menu popping up. With multicast connections there is also an option "Unprotect all connections of this multicast group"

Specifying end point devices explicitly

When starting a connection as described above, the software automatically selects the first available end point device at the selected location. You can explicitly specify the end point

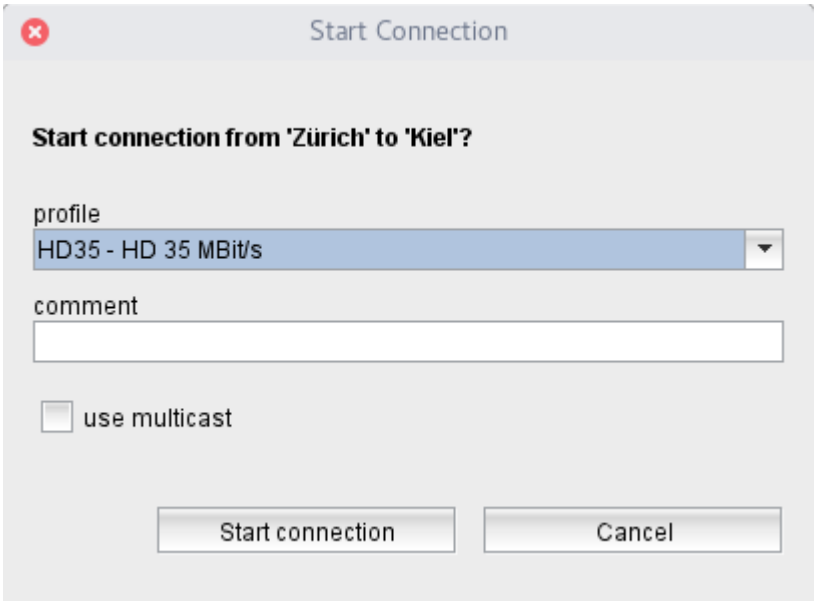
device to be used for a connection.

For this instead of left-clicking the location button use the right mouse button to show the context menu of the location button. Select the end point device of your choice from this menu.

Using a non-standard video profile / data rate

Connection which are established with the button "Activate selection" all use the default video profile and the bandwidth associated with this profile. The default profile is the first profile in the list of video profiles.

Using the button "Activate selection (user defined)" instead opens a dialog which lets you select a video profile from a list. This dialog also permits to set the "comment" property of a connection directly when the connection gets started.



Starting a multicast connection to a single destination

Using the "Activate selection (user defined)" button and the dialog window appearing with a click to this button, you may also start a connection to a single destination in multicast mode. To do this, set the check mark "use multicast" before clicking "Start connection".

Other functions

You may check the route, the software calculates for connections between two locations: Select a source location in the TX button array and then right-click the destination in the RX button array and select "Show route to here". The program displays a pop up window with the route.

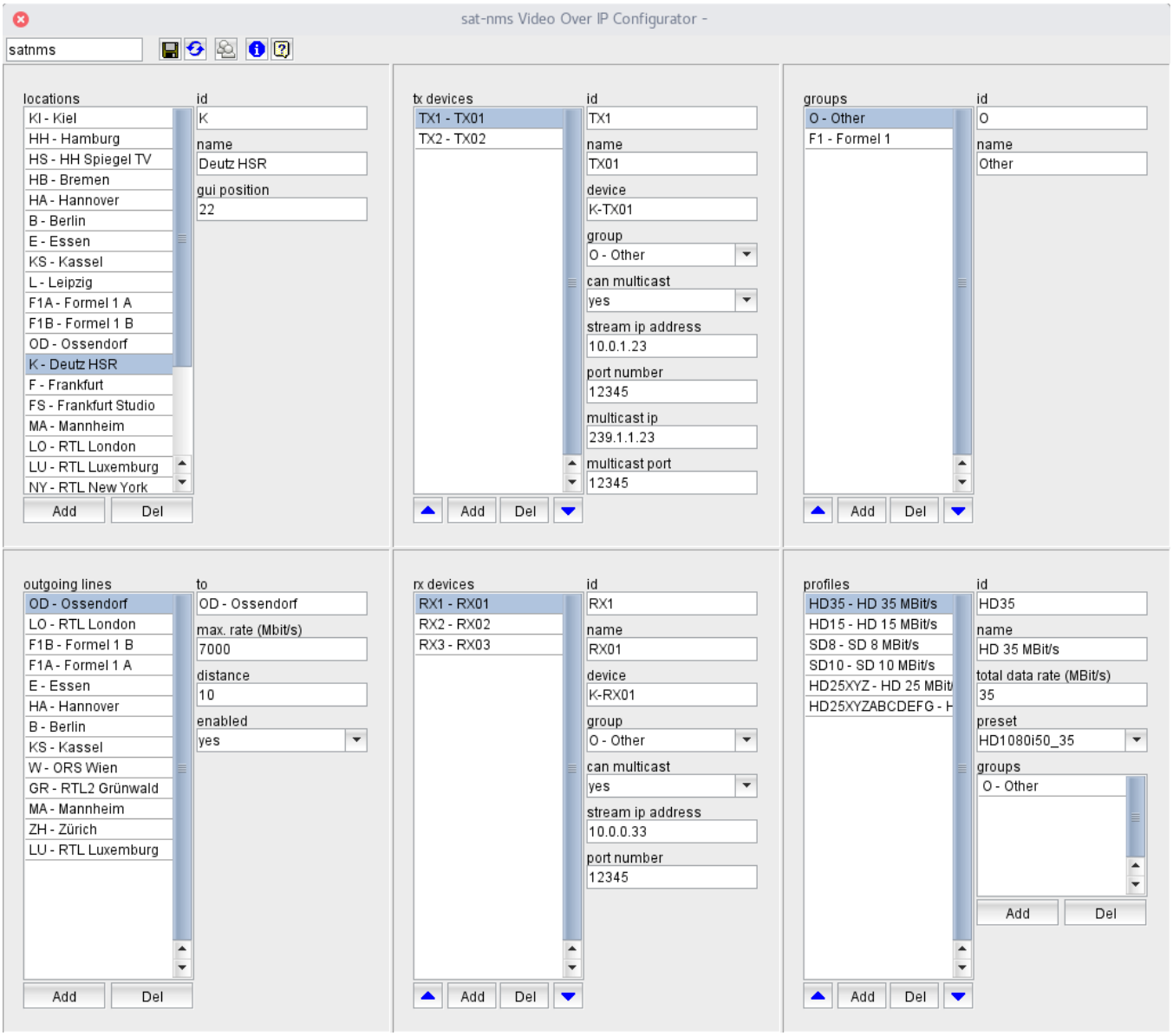
You may check the devices and outgoing lines for a location: Right-click the location button and select "Show location properties". The program displays a pop up window with requested information.

3.15.4 VioIP Network Configuration Application

The VioIP Network Configuration Application lets you configure many aspects of the VioIP Network. You can add or delete locations, endpoint devices or data lines between the locations.

The application works as a document editor: when started, it reads the actual VioIP network configuration and shows the settings. Edited settings are applied when the configuration is saved by clicking the "save" button in the tool bar. At any time you can revert the configuration to the settings which are actually running, e.g. if you deleted accidentally a location which shall not be deleted.

When you save an edited configuration, the program informs all running instances of the VioIP Client Application to reload the configuration in order to reflect the changes immediately. This is done unless you added or removed devices, in this case a restart of the M&C server is necessary to make the changes effective.



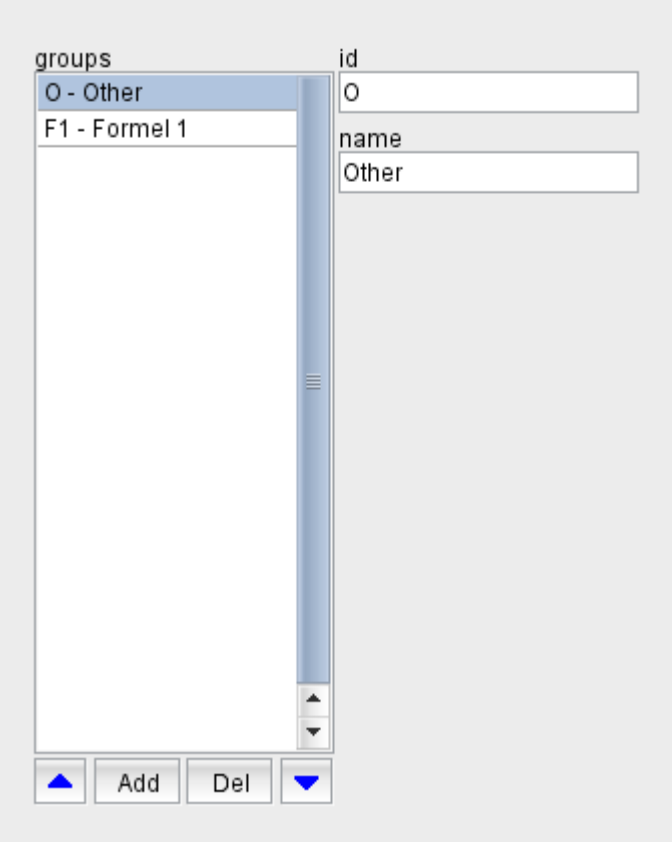
The configuration editor window is divided into 6 sections:Locations , Outgoing Lines , TX Devices , RX Devices , Groups and Profiles . While the last two sections work independently from the others, the sections Outgoing Lines , TX Devices and RX Devices always refer to the location actually selected in the Locations section.

The tool bar at the top of the window gives access to a number of global functions in the software. Below there is a short description for each of the toolbar buttons.

- --- The left field at the window's tool-bar shows the login-name of the user who is actually logged in. You may click with the mouse to this field, the program pops up the login/logout dialog. Please note, that you need a privilege level of 150 or higher to change the configuration of the software
- --- Writes the edited configuration back to the server. If you added or removed end points / devices this also changes the M&C device setup. In such a case a restart of the M&C server is necessary to make the changes effective. The software gives you the option to restart immediately or to postpone the restart.
- --- Reverts all changes.te .ts Opens the User Management Window .
- --- Shows the program version info.
- --- Shows this help page.

3.15.4.1 Groups

The sat-nms VioIP software lets you define groups of operators which only have access to a well defined part of the equipment. The VioIP Client Application is always started for one particular group and shows only the resources assigned to this group.



The group definition section is located in the top right corner of the application window. It shows the list of defined groups, selecting a group in the list shows the properties of this group right beside the list. You may edit the name of the selected group to your needs, the group name acts as a comment only. The group is uniquely identified by its ID, this cannot be changed once a group has been defined.

With the blue arrow buttons below the list, the selected group can be moved up and down in the list. The position of a group in the list is important inasmuch as the topmost group in the list is considered to be the default group.

The button "Add" lets you add a new group. The program shows a dialog window where you can enter the ID for the new group. An ID entirely consists of upper case letters and digits, the "-" character is allowed as separator. The ID must be unique, you cannot define two groups with the same ID. The ID also is unalterable, once you defined a new group you cannot change the ID of the group later.

The button "Del" lets you delete the selected group. The program shows a query dialog before the group actually gets deleted. The group you want to delete will probably be referenced by other resources, with these the following will happen:

- End point devices which belong to the deleted group will be moved to the default group. If you are going to delete the default (the topmost) group, the devices are moved to the group which will be the default group after the deletion. The very last group cannot be deleted. The software shows a second query dialog stating something like "9 devices will be moved from group 'A' to group 'B'" before the group deletion actually is done to inform

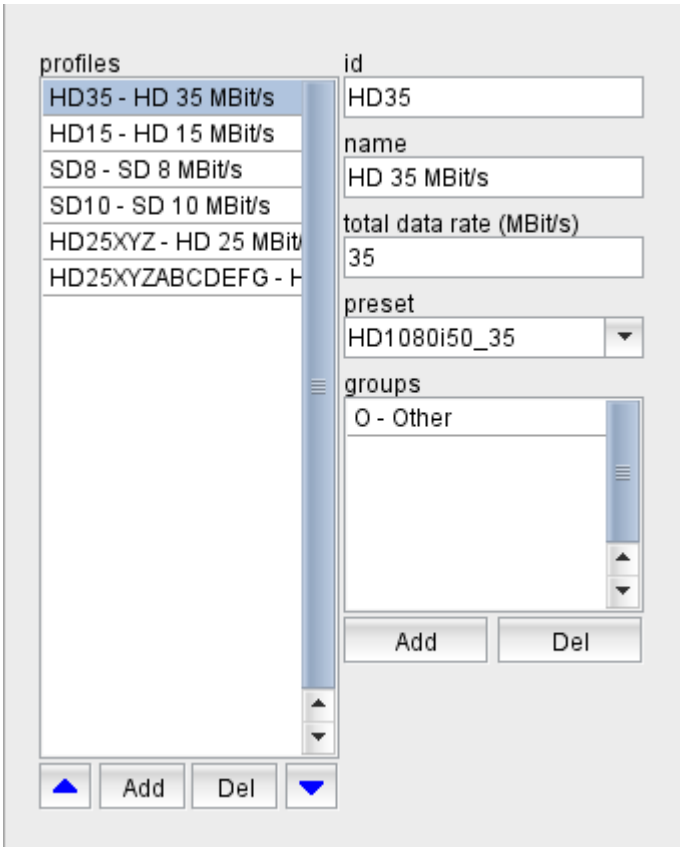
you what will happen if you delete this group.
- The group is removed from all profiles which had this group attached.

3.15.4.2 Profiles

Profiles define the data rate of the encoded stream, the video and audio encoding of the signal and many more parameters. The software uses a profile by applying a preset to the devices representing the end points of the upcoming connection.

To make a profile universally usable, presets with the name defined in the profile must be present for all types of encoder/decoder equipment used by the software. The software checks this when a new profile is defined.

There are device types which contain multiple encoders or decoders in one device frame. The software knows about these device types and expects a preset "NAME" to be present as "NAME-1", "NAME-2" ... "NAME-n" for all encoder or decoder instances this device may contain.



The profile definition section is located in the bottom right corner of the window. It shows the list of defined profiles, selecting a profile in the list shows the properties of this profile right beside the list. You may edit the properties of the selected profile to your needs. The profile is uniquely identified by its ID, this cannot be changed once a profile has been defined.

- *id:* The profile ID internally identifies the profile. It cannot be changed once the profile has been created.
- *name:* The profile name is shown in the VioIP Client Application in the list of active

connection and it is used to select a profile when the operator starts a connection with the "Activate selection (user defined)" button.

- *total data rate (MBit/s):* In this field enter the total data rate the encoded signal occupies on the data lines between the locations. The rate must match the data rate produced by the encoder preset and must include the rate of all contained streams including the transport stream format overhead. This is **not** the video data rate!.
- *preset:* With this field the name of the preset to be used for this profile is defined. The software presents a drop down list with all presets it finds for the device types known to be used as encoders / decoders within the VioIP software. The entry field also permits to overwrite the name and set a preset name which is not yet present. When the preset name is changed - either by entering a new name or by selecting a name from the list - the software checks if this preset is available for all known encoder / decoder device types. If not the software displays a warning. This means, you can define a profile which has the referenced presets not or not completely defined, but you are warned: You will not be able to start a connection with this profile if the referenced preset is not available for either the source or the destination device.
- *groups:* A VioIP Client Application will only permit to use this profile if the group ID the client has been started with appears in this list. This way you may define different profile settings for each particular group, each group will only see the profile definitions it has been granted permission. You may use the "Add" / "Del" buttons below the groups list to add or delete groups for this profile. If the group list of a profile is empty, it cannot be used.

With the blue arrow buttons below the profile list the selected profile can be moved up and down in the list. The topmost profile in the list is considered to be the default which is used for connections the operator starts without the "user defined" option.

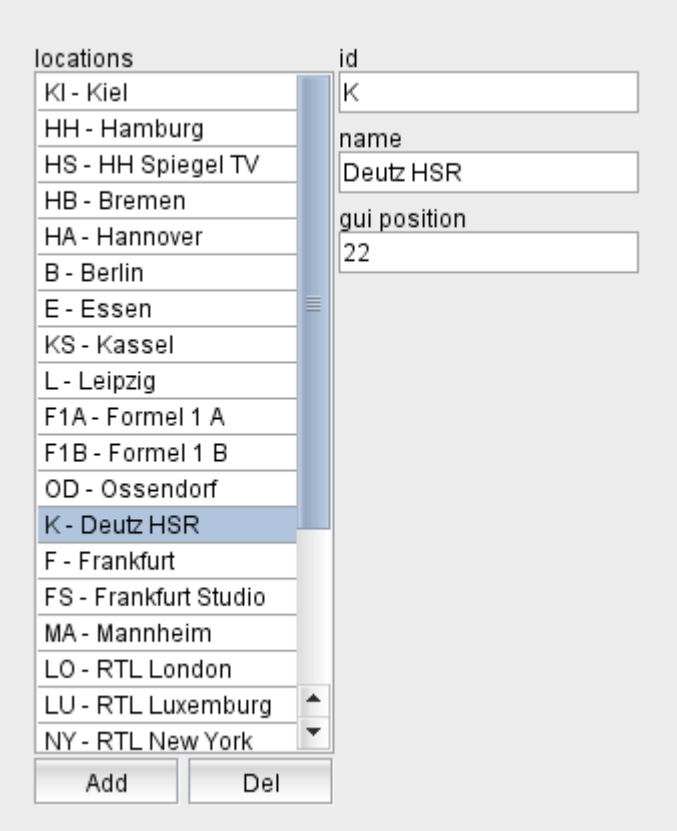
The button "Add" lets you add a new profile. The program shows a dialog window where you can enter the ID for the new profile. An ID entirely consists of upper case letters and digits, the "-" character is allowed as a separator. The ID must be unique, you cannot define two profiles with the same ID. The ID also is unalterable, once you defined a new profile you cannot change the ID of the profile later.

The button "Del" lets you delete the selected profile. The program shows a query dialog before the profile actually gets deleted. Deleting a profile does **not** delete the preset definitions referenced by this profile.

3.15.4.3 Locations

The section in the top left corner of the window shows the list of locations. Locations typically match rooms or cities, but strictly spoken 'locations' in the sense of this software are logically places which share a number of encoder/decoder devices which are connected in a network with virtually infinite bandwidth (i.e. the bandwidth of the network between the local devices is not managed by the software).

A location has a number of outgoing data lines and a number of TX and/or RX devices attached.



The 'locations' section shows the list of defined locations, selecting a location in the list shows the properties of this location right beside the list. Selecting a location also updates the data in the Outgoing Lines , TX Device and RX Devices sections to show these objects as they are available at the selected location.

You may edit the properties of the selected location to your needs. The location is uniquely identified by its ID, this cannot be changed once a location has been defined.

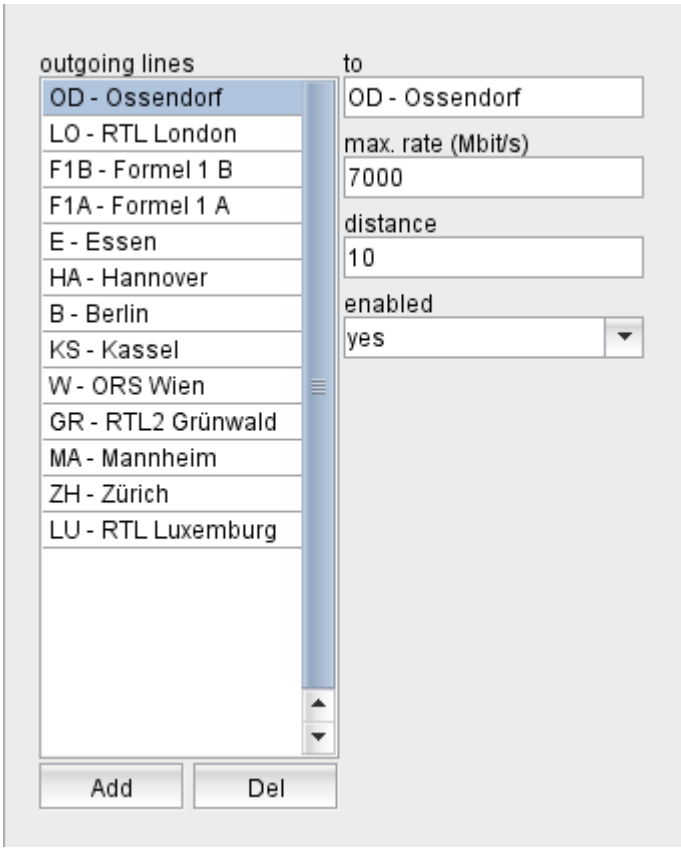
- *id:* The location ID internally identifies the location. It cannot be changed once the location has been created.
- *name:* The location name is shown in theVioIP Client Application as the label of the button representing the location in the TX/RX button arrays.
- *gui position:* The value in this field defines the position of the button representing this location in the button arrays. These arrays consist of 7 rows / 5 columns. A GUI position "1" addresses the button in the upper left corner, "5" in the upper right corner. "6" addresses the first button in the second row and so on. The special value "0" makes a location (temporarily) invisible. You cannot assign a GUI position twice, the program will decline this. If a new location is defined it automatically gets the first available GUI position assigned as a default.

The button "Add" lets you add a new location. The program shows a dialog window where you can enter the ID for the new location. An ID entirely consists of upper case letters and digits, the "-" character is allowed as a separator. The ID must be unique, you cannot define two locations with the same ID. The ID also is unalterable, once you defined a new location you cannot change the ID of the location later.

The button "Del" lets you delete the selected location. The program shows a query dialog before the location actually gets deleted. Deleting a location does **not** delete any sat-nms device definitions associated with this location.

3.15.4.4 Outgoing Lines

The section "outgoing lines" below the "locations" section defines the data lines which go from the selected location to other locations. The software manages data lines as unidirectional connections from location A to B. To define bidirectional lines, you have to define the line twice: once as a line from A to B and a second time as a line from B to A.



The section "outgoing lines" shows the list of data lines starting at the actually selected location. Selecting a line in the list shows the properties of this line right beside the list.You may edit the properties of the selected data line to your needs

- *to:* The ID and name of the destination location of this line. This field is read-only and cannot be changed.
- *max. rate (Mbit/s):* The maximum data rate which is available for the VioIP application on this line. The software uses this to examine if there is enough bandwidth available for a new connection.
- *distance:* The virtual distance between the two locations connected by this line. The software uses this number to calculate the data route between two locations using the Dijkstra algorithm. For details see the paragraph at the end of this page.
- *enabled:* This switch lets you temporarily disable a line without removing the line from

the configuration.

The button "Add" lets you add a new line. The program shows a dialog window where you can select the destination location for the line.

The button "Del" lets you delete the selected line. The program shows a query dialog before the line actually gets deleted.

Route Calculation

In the VioIP system the route from one location to another is determined by the settings of the IP routers in the system and by source/destination IP addresses. To calculate the available bandwidth between two locations, the software must know the way the IP packets take, specially when the route goes over several hops. The available bandwidth then is calulated as the minimum bandwidth available on any line on this route.

The software uses for this the "shortest path" algorithm by Dijkstra: Each line has assigned a "distance" value, the total distance of a route is the sum of distances of all hops of the route. From all possible routes the software chooses the one with the shortest total distance.

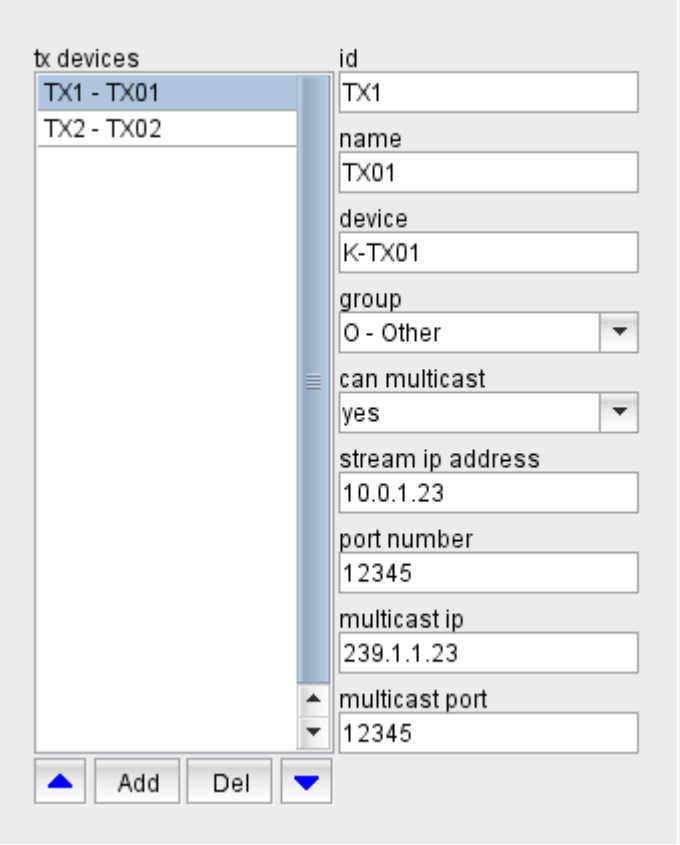
As the software cannot read the settings of the routers used in the network, it is important to set the "distance" parameters in a way that the routes calculated by the VioIP software match the routing programmed in the IP routers. You may check the routes actually used in the software in the VioIP Client Application : Select a source location in the TX button array and then right-click the destination in the RX button array and select "Show route to here". The program displays a pop up window with the route.

3.15.4.5 TX Devices

The section "tx devices" manages the transmit devices or more strictly spoken the transmit side connection end points available for the selected location. This usually comprises more than the plain encoder device.

A transmit side connection end point consists of the following components:

1. The data stored about this end point in the VioIP configuration database.
2. A logical end point (VioIP-End-Point) device representing the connection end point in the sat-nms M&C software. The logical end point device receives abstract commands to start or stop a connection and translates this abstract command into the particular settings expected by the physical end point device. There is always exactly one logical end point device for a connection end point.
3. A physical end point device in the sat-nms M&C software. This device - typically an encoder - gets the video signal at its input and creates an IP stream sent over the network. There are devices containing more than one encoder unit in one physical device frame. Hence it is possible that multiple logical endpoint devices share the same physical device. A so called "module index" is used to identify the encoder unit which shall be used.
4. Optionally there may be a file recorder device assigned to the connection end point. This file recorder logs quality parameters of the physical endpoint device like packet counters.



The section "tx devices" shows the list of tx devices available at the selected location. Selecting a device in the list shows the properties of this device right beside the list. You may edit the properties of the selected device to your needs

- *id:* The device ID internally identifies the device. It cannot be changed once the device has been created.
- *name:* The name of the device.
- *device:* The name of the VioIP-End-Point device representing this end point in the sat-nms M&C software.
- *group:* The group this device is assigned to. The device is only visible / accessible for a VioIP Client Application started for the ID of this group .
- *can multicast:* Set this to "yes" if the physical device representing this end point is capable to create a multicast stream.
- *stream ip address:* Enter the IP address of the stream port of the device here. This is the source address of unicast streams sent by this device. The address usually is not actively commanded at the device, however the value entered here must match the address configured in the device.
- *port number:* This is the port number of the source of a unicast stream. Usually this value doesn't matter as the device uses a random port number for this.
- *multicast ip:* The multicast address / group to be used in multicast mode.
- *multicast port:* The port number to be used in multicast mode.

With the blue arrow buttons below the device list the selected device can be moved up and down in the list. When the software searches for the first available deviceThe topmost device

which is used for connections the operator starts without specifying a device explicitly, it goes through this list from top to bottom.

The button "Add" lets you add a new device. The program shows a dialog window where you can enter the ID for the new device. An ID entirely consists of upper case letters and digits, the "-" character is allowed as a separator. The ID must be unique, you cannot define two devices with the same ID. The ID also is unalterable, once you defined a new device you cannot change the ID of the device later.

When you entered the ID for new device, the software shows in a next step a dialog which lets you select which aspects of the connection end point actually shall be created and how these aspects shall be initialized. Chapter Adding New Devices later in this document describes this dialog more elaborately.

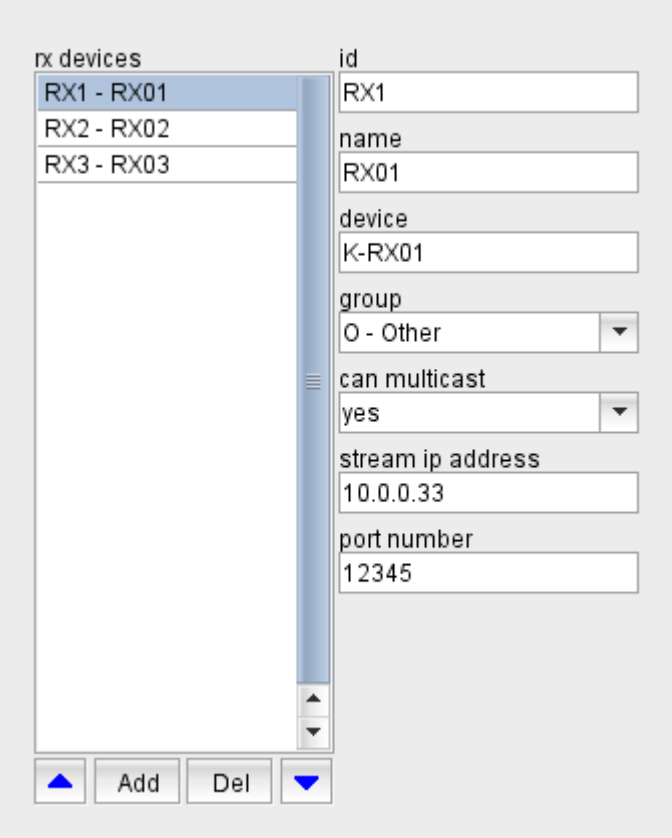
The button "Del" lets you delete the selected device. The program shows a query dialog before the device actually gets deleted. The dialog lets you select if the software shall remove the VioIP-End-Point device, the physical device and an optional file recorder device as well from the M&C system configuration.

3.15.4.6 RX Devices

The section "rx devices" manages the receive devices or more strictly spoken the receive side connection end points available for the selected location. This usually comprises more than the plain decoder device.

A receive side connection end point consists of the following components:

1. The data stored about this end point in the VioIP configuration database.
2. A logical end point (VioIP-End-Point) device representing the connection end point in the sat-nms M&C software. The logical end point device receives abstract commands to start or stop a connection and translates this abstract command into the particular settings expected by the physical end point device. There is always exactly one logical end point device for a connection end point.
3. A physical end point device in the sat-nms M&C software. This device - typically a decoder - receives the IP stream at its input and decodes it to the contained video / audio signals. There are devices containing more than one decoder unit in one physical device frame. Hence it is possible that multiple logical enpoint devices share the same physical device. A so called "module index" is used to identify the decoder unit which shall be used.
4. Optionally there may be a file recorder device assigned to the connection end point. This file recorder logs quality parameters of the physical endpoint device like packet counters.



The section "rx devices" shows the list of rx devices available at the selected location. Selecting a device in the list shows the properties of this device right beside the list. You may edit the properties of the selected device to your needs

- *id:* The device ID internally identifies the device. It cannot be changed once the device has been created.
- *name:* The name of the device.
- *device:* The name of the VioIP-End-Point device representing this device or end point in the sat-nms M&C software.
- *group:* The group this device is assigned to. The device is only visible / accessible for a VioIP Client Application started for the ID of this group .
- *can multicast:* Set this to "yes" if the physical device representing this end point is capable to receive a multicast stream.
- *stream ip address:* Enter the IP address of the stream port of the device here. This is used as the destination address of unicast streams sent to this device. The address usually is not actively commanded at the device, however the value entered here must match the address configured in the device.
- *port number:* This is the destination port number for the unicast stream.

With the blue arrow buttons below the device list the selected device can be moved up and down in the list. When the software searches for the first available deviceThe topmost device which is used for connections the operator starts without specifying a device explicitly, it goes through this list from top to bottom.

The button "Add" lets you add a new device. The program shows a dialog window where you can enter the ID for the new device. An ID entirely consists of upper case letters and digits, the "-" character is allowed as a separator. The ID must be unique, you cannot define two devices with the same ID. The ID also is unalterable, once you defined a new device you cannot change the ID of the device later.

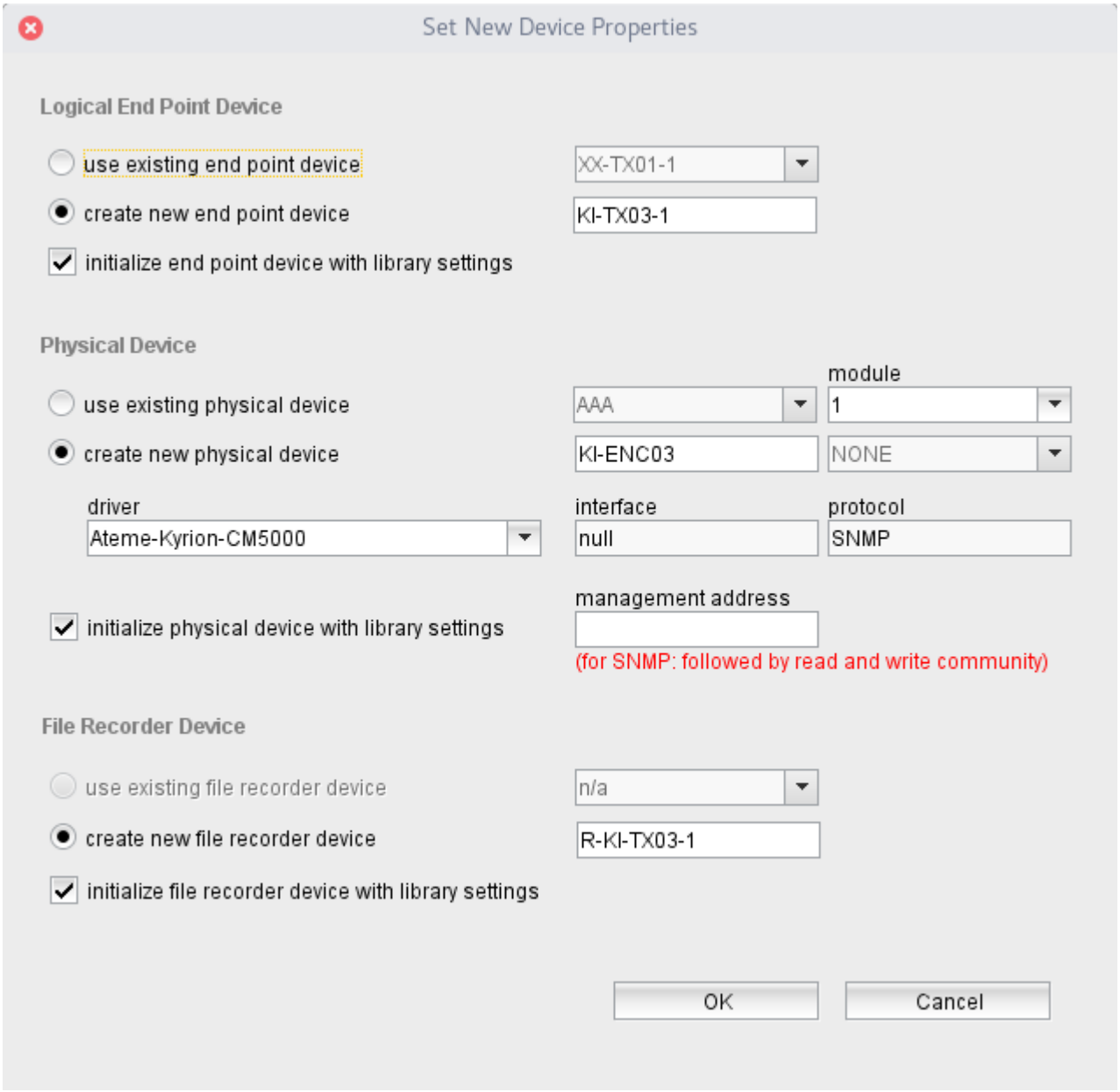
When you entered the ID for new device, the software shows in a next step a dialog which lets you select which aspects of the connection end point actually shall be created ans how these aspects shall be initialized. Chapter Adding New Devices later in this document describes this dialog more elaborately.

The button "Del" lets you delete the selected device. The program shows a query dialog before the device actually gets deleted. The dialog lets you select if the software shall remove the VioIP-End-Point device, the physical device and an optional file recorder device as well from the M&C system configuration.

3.15.4.7 Adding New Devices

The button "Add" below the list of TX or RX devices lets you add a new device. The program shows a dialog window where you can enter the ID for the new device. An ID entirely consists of upper case letters and digits, the "-" character is allowed as a separator. The ID must be unique, you cannot define two devices with the same ID. The ID also is unalterable, once you defined a new device you cannot change the ID of the device later.

When you entered the ID for new device, the software shows in a next step this dialog which lets you select which aspects of the connection end point actually shall be created and how these aspects shall be initialized.



The first section in this dialog refers to the logical end point device which represents the new connection end point to create. You have the choice either to use an existing VioIP-End-Point device in the M&C for this or to create a new one.

You can select the first choice only if there areVioIP-End-Point devices in the M&C which are not yet used for the VioIP software. The software offers a list of such devices in this case, you can select one from this drop down list.

If you select to create a new logical end point device, the software suggests a name for this device, made up from the location ID and the device ID you specified in the previous step.

Finally the checkmark "initialize end point with library settings" defines if the setup of the used or newly created device shall be initialized with settings derived from choices made in the next step concerning the physical device. This includes the faults to be monitored at the physical

device and the start/stop macros used by the logical end point device to perform the start and stop of a connection.

The next section is all about the physical device to be used for the end point. Again you have the choice either to use an existing device or to create a new one. Using an existing device is speically useful to use an unused module in a multi-encoder or multi-decoder device. Regardless of your choice you have to specify a module index beside the device name if the device containes multiple modules.

If you decide to create a new physical device you should select the device driver for the new device as the next step. Selecting the driver sets the communication interface and communication protocol defaults and suggests a name suitable for the new device.

The next step is to specify the management address for the new device. For devices controlled via SNMP the address should contain also the read and write community names (e.g. "10.0.0.1 public private").

The checkmark "initialize physical device with library settings" defines if the setup of the used or newly created device shall be initialized with settings derived from choices made in the next step concerning the physical device. This is important for multi-encoder or multi-decoder devices because this masks stream specific faults of the device in a way, that these faults do not interfere with the state of other streams on the same device.

The last section of the dialog defines the optional file recorder which may be used to log the values of packet counters of the physical device. Like the logical end point device, the file recorder is a logical device and the options given for this device are very similar to the options for the logical end point.

You may choose between using an existing device (if one is available) or to create a new one. Checking "initialize file recorder with library settings" lets you automatically initialize the file recorder to monitor the packet counter parameters of the physical device selected above.

3.16 M&C Configuration

This section of the manual describes how to configure the sat-nms M&C software for the device configuration of the ground station to control and how to create an application specific user interface consisting of one or more screens.

The following chapters refer to the legacy Java client user interface of the software, they only are applicable to software installation which still or explicitly use the Java client as user interface.

New M&C installation should preferably use the Web based user interface which provides equivalent funtions to configure the software for a specific task.

3.16.1 Configuring The Device Setup

With the sat-nms M&C or VLC software, the assignment which type of device is controlled by which serial interface is free configurable. The user interface of the software provides a

configuration window which enables the authorized operator to setup the software.

Chapter ' The Device Setup Editor Window ' describes the usage of this configuration tool and guides through setup of a M&C system or VLC. The now following chapter introduces the underlying concepts of the M&C / VLC device configuration.

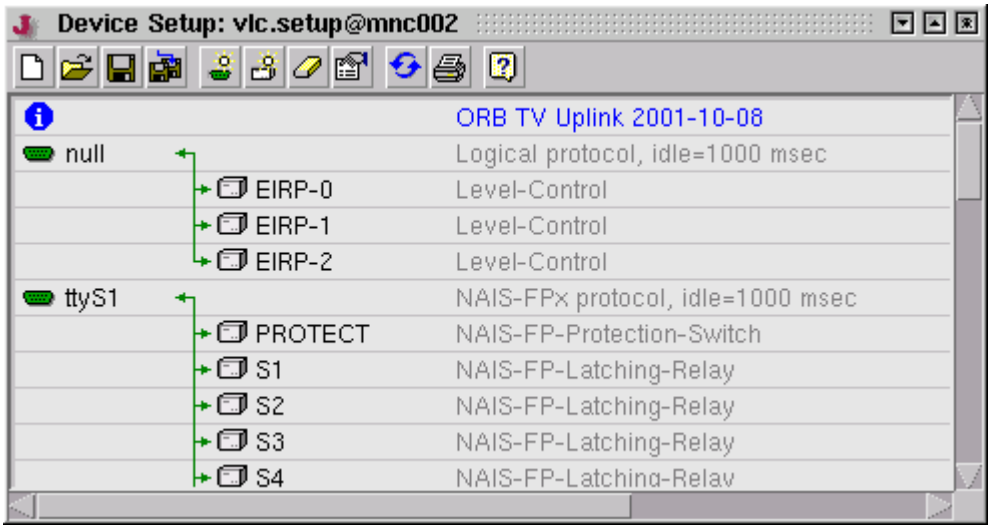
3.16.1.1 Concepts

When an operator is going to configure a VLC or M&C first it is important to understand, that with the sat-nms software each serial interface is bound to a communication protocol. The communication protocol used for a certain device is not unchangeably linked to the device driver. This is to support multiple devices being controlled via a single interface. Devices using a common 'multi drop' capable protocol may be connected to the VLC / M&C with a RS485 bus. This way one can for example operate a ND-SatCom upconverter and a CPI TWTA with a single RS485 interface. Both device types use the same communication protocol, they may share the same RS485 bus.

Once an interface (serial port) has been included into the setup and got a communication protocol assigned to it, within the setup devices can be attached to this interface. With the device definition, the device gets a name and a device driver assigned to it. The device's name acts as an address within the software where to send at commands (equipment settings) to, the device driver tells the software how to treat the device at a low communication level.

3.16.1.2 The Device Setup Editor Window

The 'Device Setup Editor' lets you define / modify the equipments setup, i.e. the assignment of monitored/controlled devices to the physical interfaces of a VLC or M&C computer. You need to be connected on-line and logged in as a fully authorized operator to edit the equipment setup.



As shown in the figure above, the editor presents the equipment setup as a list of interface and device definitions. Green arrows connect the devices with the interface they use to communicate with the computer.

In the VLC/M&C computer, the equipment setup is stored as a file 'vlc.setup'. This file is read by the M&C server software on startup. Changes you make in the editor window are temporary until save this file to the target system (the function). A changed equipment setup you stored on the target machine will become effective when the M&C server application gets restarted.

Tool-bar Functions

- --- Clears the setup, lets you define a new setup from scratch.
- --- Loads a equipment setup from disc. With a NMS system the file is loaded from the NMS server, not from the VLC.
- --- Saves the edited setup as 'vlc.setup' to the target computer. It will use this equipment setup with the next start.
- --- Saves the edited setup with a different file name. Setup definitions are stored in a folder 'setups' which is located at the NMS server computer with NMS systems, on the M&C server computer with M&C systems.
- --- Adds a new interface definition. The definition is inserted just below the actually highlighted one.
- --- Adds a new device definition to the setup. The new definition is inserted just below the actually highlighted one unless the latter describes an interface. In this case, the new definition gets appended at the end of the list of devices already connected to this interface
- --- Deletes the actually highlighted definition.
- --- Lets you edit/change the actually highlighted definition. You also may double click to the definition for this.
- --- Restarts the M&C server application on the target computer. You must do this, to make the target system use a modified equipment setup. With a M&C application, this will break the user interface's connection to the server. With a NMS/VLC application the VLC will go off-line with the restart. You should wait at least 2 minutes before you try to connect again.
- --- Prints the edited setup.

Multiple Devices At One Interface

With RS232 interfaces, you will connect one piece of equipment to one interface in most cases. However, there are a couple of situations where multiple devices at one interface make sense:

- The *null* interface is used by all logical devices. In every VLC/M&C system there is exactly one *null* interface running the *Logical* protocol. All logical devices are connected to this (physically not existent) interface.
- When using an RS422/485 interface, you may connect several devices to a bus and control them through one interface. The precondition for this is, that **all** devices at this interface use the **same** communication protocol supporting this kind of multi drop operation.
- Devices for switches, alarm circuits etc. which rely on the Matsushita FP0 PLD, are assigned to the same interface as long as they are implemented by the same FP0 device.

Storing Setups For Backup Or As Templates

With an NMS application you are encouraged to backup every equipment setup at the NMS site by means of the 'save as ( )' function. This backs up the settings at a central place and builds a set of template setups.

Restarting The Target System

When modifying an equipment setup, be sure to store the changes setup using the - function and then to restart the target with . Without restart, the target continues using the old setup until a restart occurs for some other reason.

Stored Equipment Settings

Apart from the plain interface assignments done by this configuration window, the equipment setup comprises a large number of settings like interface line speeds, device addresses etc. When changing device names, be aware that these settings are lost.

3.16.2 Arranging The User Interface

The sat-nms software contains a powerful tool which enables the operator to customize or even create user interface screens which are nice looking and easy to operate. The tool, called the ' Screen Editor ', works much like a vector based drawing program. It lets the user place various kinds of objects (editing fields for settings, textual or graphical status displays and much more) into the screen to edit.

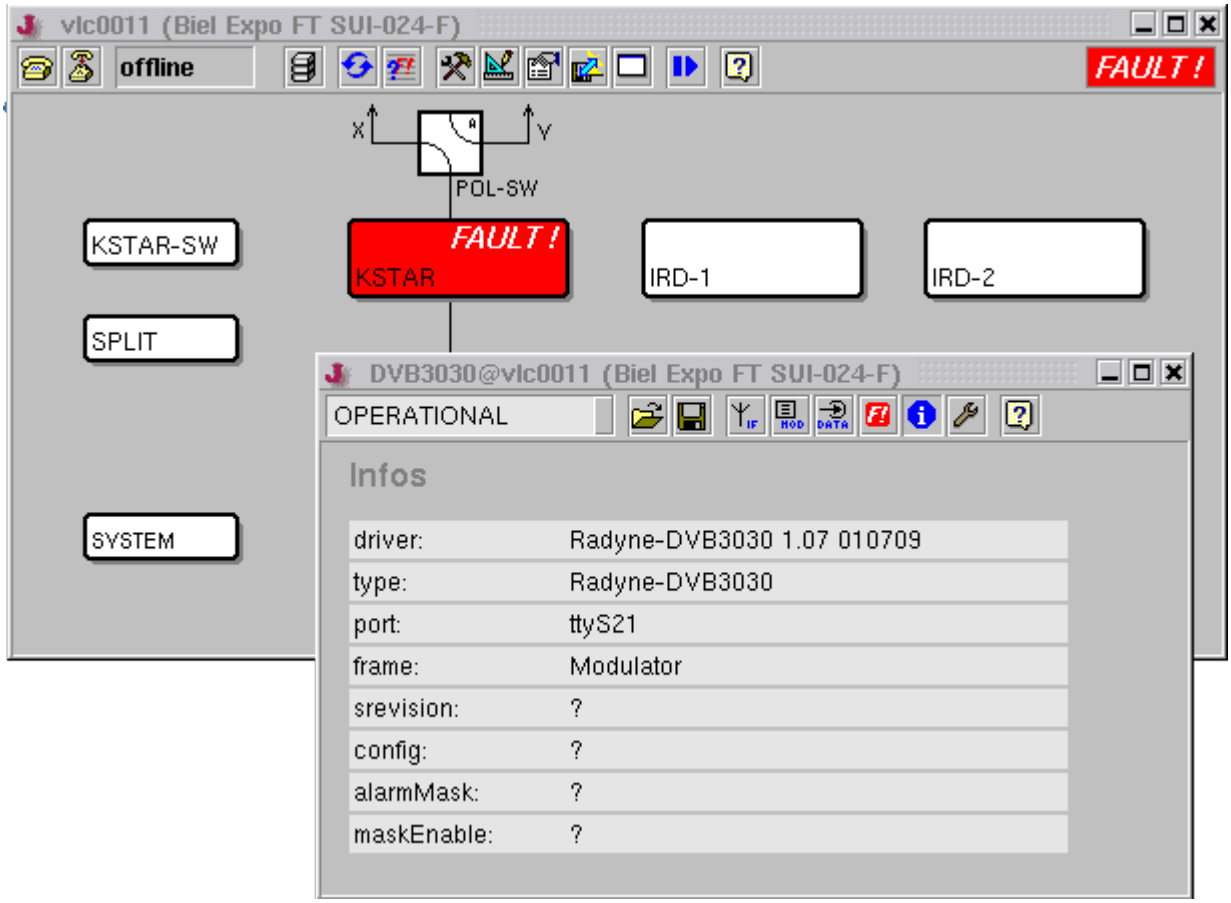
The software comes with predefined screens for each device type. These screens can be used to create a hierarchical structured user interface with a few mouse clicks. On the other hand, the configuration capabilities of the sat-nms user interface are powerful enough to create special user interface screens exactly matching the operator's needs.

3.16.2.1 Concepts

The software foresees one main screen for each M&C / VLC. With a M&C this main screen automatically is shown when the user interface program is started. With a VLC, the 'VLC Main Window' contains this screen. The user interface may be configured to use more than one screen which are switched on the operator's request, either shown in the same window or in separate windows.

Device oriented user interface

There are two major approaches to design and configure a user interface for the sat-nms software. The first one, called the 'device oriented user interface', results in a hierarchically structured user interface. It makes use of the predefined windows / screens coming with the software and can be created with only a few mouse clicks.



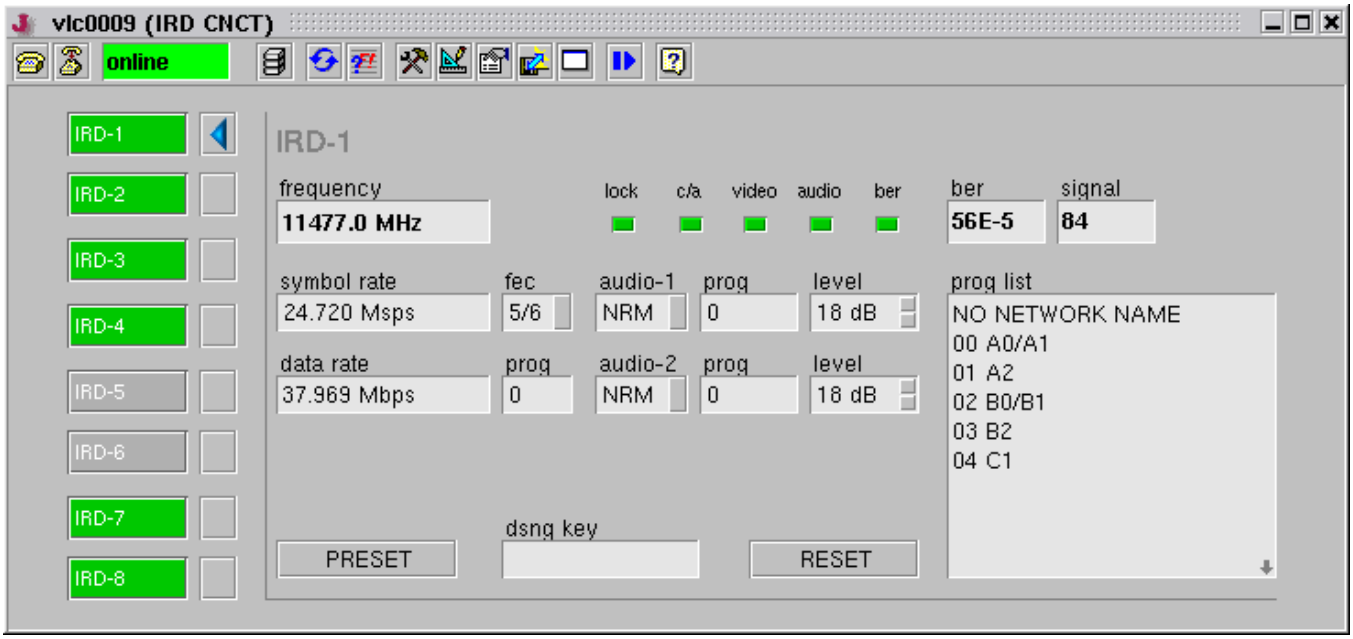
Example for a device oriented user interface.

With the device oriented user interface, the main window contains a so called device icon for each device which is controlled by the M&C/VLC. This icon is labeled with the device's name and shows the OK/FAULT state of the device with it's color. A double click to a device icon opens the ' Device Window ' for this device, containing the predefined screens for this device type. Using the ' Device Window ', all parameters of a device supported by the sat-nms software can be monitored or controlled.

To create a main screen for the device oriented user interface, thescreen editor provides a function adding a device icon for each device in the system on a single mouse click.

Task oriented user interface

While the device oriented user interface follows predetermined structures, the so called 'task oriented user interface' is entirely customer designed. Using the screen editor , the user places entry fields or status displays in the screen for each parameter he wants to include to the user interface.



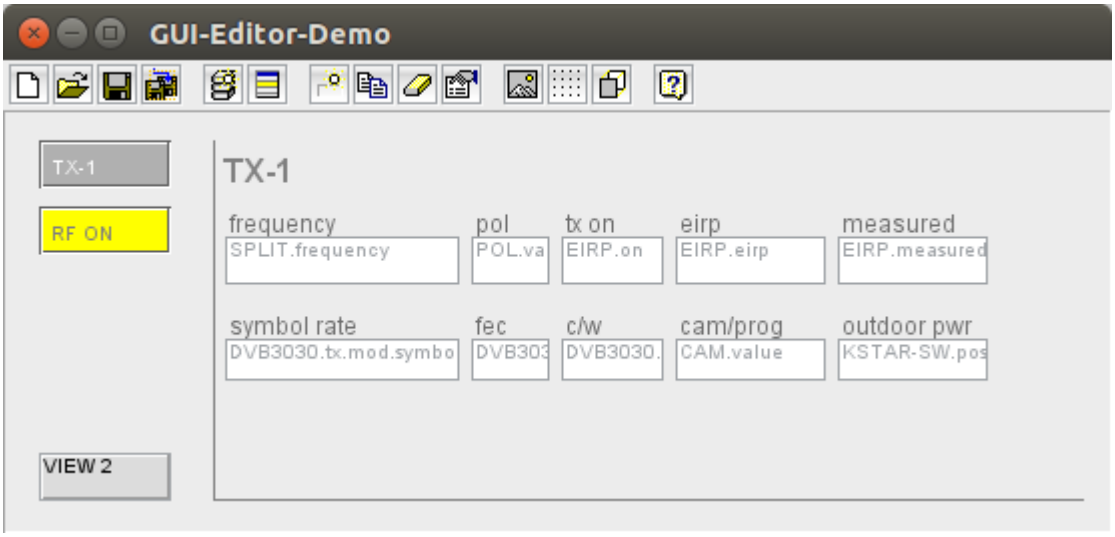
Example for a task oriented user interface.

A task oriented user interface not necessarily contains all available parameters. To keep the user interface clear, only those parameters are included, which the operator needs to fulfill the task the user interface is designed for. Moreover, there also is no need to group parameters by the device they belong to, like the device oriented user interface does. Parameters belonging to diverse devices freely may be mixed in one screen.

3.16.2.2 The Screen Editor Window

The Screen Editor window is used to layout the M&C windows used in the M&C/NMS user interface. Windows like the VLC Main window, the M&C Main window and additional user created windows are customized by means of this program.

The Screen Editor works like a simple drawing program. Each M&C window contains a number of objects (elements) which may be placed, sized and edited using the Screen Editor.



Object Placement

Objects contained in the window are placed by dragging them to the desired position (left mouse button). Dragging an object with the right mouse button changes it's size, or, more precisely changes the size of it's bounding rectangle (most objects are not scaled if they are resized). A double click to an object opens the Element Properties window for this object.

Tool-bar Functions

- --- Clears the window surface and starts to edit a new window.
- --- Loads a window description to edit.
- --- Saves the window actually edited.
- --- Saves the window actually edited with a different name. If you have selected a group of elements, only the selected elements are saved. This deletes all unselected elements and shrinks the panel just to fit around the selected elements! The saved screen always ist stored in the "/screens" directory. This applies even if you edited a VLC screen from within the NMS client application.
- --- Adds a new device icon for each device which has not yet an icon. This function is only available, if the program knows about the device list of the application which is edited.
- --- Adds a new object to the window. It appears in the upper left corner. You also may point with the mouse to the (empty) destination point at the drawing area and the right button context menu.
- --- Duplicates the selected object. The new object appears on top of existing one.
- --- Removes the selected object. If you have selected a group of elements, all elements in the group are removed.
- --- Opens a Element Properties window for the selected object.
- --- Defines a background image for the window. A file selection box appears, you may select any picture in GIF or JPG format as the window background. The window size automatically gets aligned to the size of the image you selected.
- --- Objects are placed along a invisible 3 x 3 pixel grid if they are moved while this

button is down.
- --- Moves the selected object to the back so it does not cover any other object.

Inserting screens

You may insert complete screens into the actually edited one. Point the mouse to where you want the upper left corner of the inserted screen to be placed. Press the right mouse button and select "Insert screen" from the context menu. A file selection box appears which lets you select the screen to insert.

You also may select one of the predefined screens used by the device windows of the software for insertion. To insert such a file, change the directory to "dscreens" in the file selection box. Be aware, that the message identifiers of all parameters in these screens don't contain a device ID. The tx-on parameter of a transceiver e.g. refers to a message ID "tx.on" in a device screen, not to something like "KSTAR-1.tx.on". You have to edit all message identifiers if you insert a screen from this directory.

Group selections

The screen editor lets you select multiple elements to a set on which you can perform some useful actions. Selecting an element group works like you would expect: Spanning a rectangle by dragging the mouse with the left button held down selects all elements which are covered by this rectangle. Selecting single elements with the mouse while pressing down the CTRL key at the keyboard also adds these element to the group. Any other mouse click clears the selection.

Elements selected in this way are marked with a blue rectangle drawn around. The first element you selected is called the "reference element". It's marked with four small boxes in it's corners. If you select a group by spanning a rectangle around the elements, the element which was selected before becomes the "reference element". Below there is a list of operations you may apply to a selected group of elements:

- Moving a group of elements: --- Select the elements you want to move, then keep the CTRL key pressed and drag one of the elements in the group (left mouse button). The other elements follow.
- Resizing a group of elements: --- You can change a group of elements to the same width, height or size the reference element has. Select the elements you want to resize and select the reference element. Then -- while pressing the CTRL key -- open the context menu with the right mouse button on one of the elements in the group. Select "Same width", "Same height" or "Same size" as you need.
- Aligning a group of elements: --- Aligning a group of elements, i.e. moving them to make either the left, right, top or bottom edge of all elements in one line, is done analogous to the resizing functions. The context menu selections are "Align left" ... "Align bottom". "Align column" is a special function which aligns the selected elements horizontally to the reference element like "Align left". Vertically this function aligns the elements to fit into a fixed 42 pixel raster, relative to the reference element.
- Re-addressing a group of elements. --- With this function (context menu selection "Same device") you can address a group of elements to the same device name the reference element contains in it's message ID. The device name in the message ID of each element

in the group gets replaced by the device name in the reference element's message ID. This function recognizes if an element contains no device name defined and adds the reference element's device name in this case.

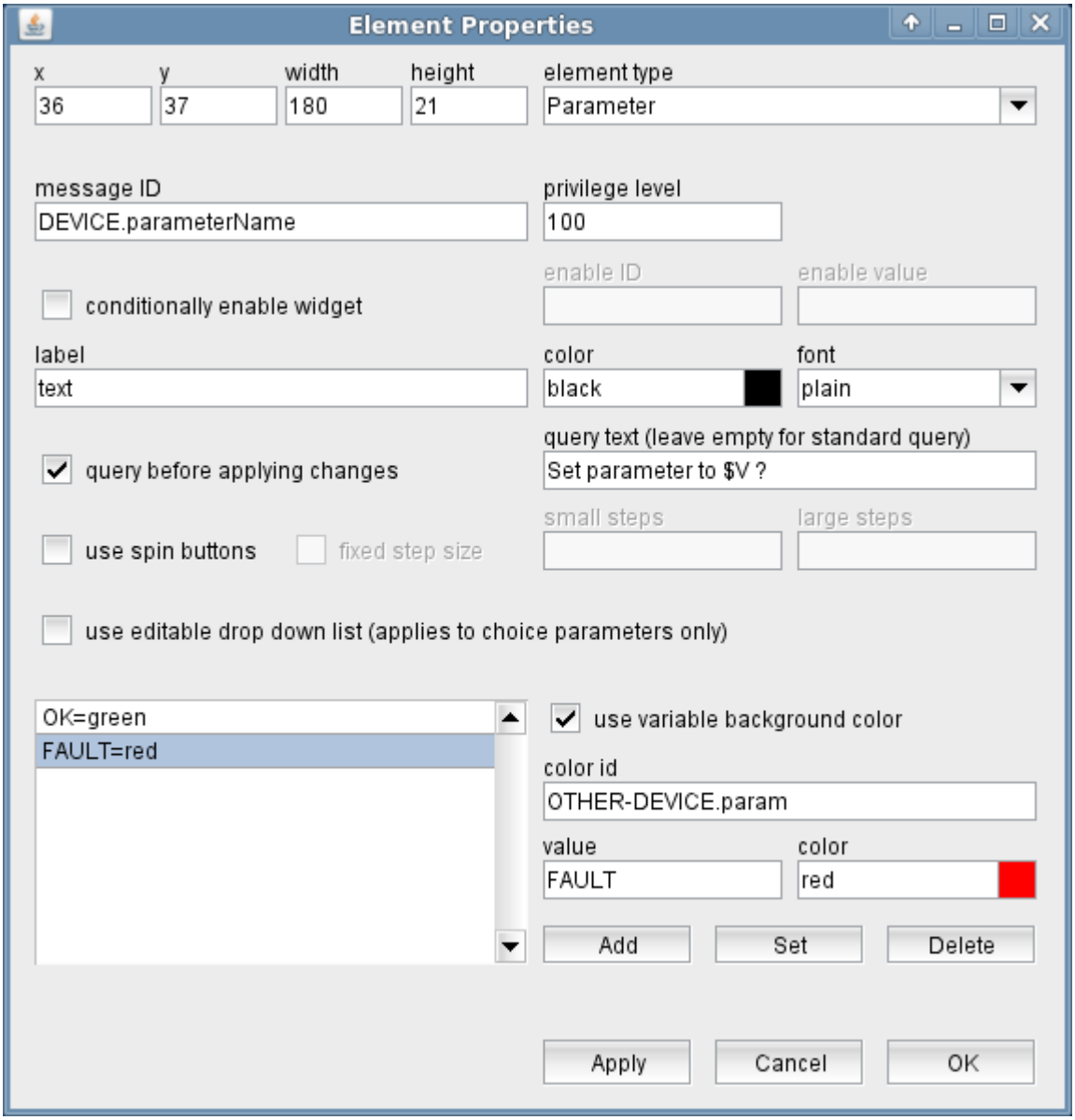
- Storing a group of elements: --- Select the elements you want to store and click to in the tool bar. This deletes all unselected elements and shrinks the panel just to fit around the selected elements, so be careful! In most cases it will be better to make a backup of the edited screen before.
- Deleting a group of elements: --- Select the elements you want to delete and click to in the tool bar.

Remarks

- Window descriptions, images and icons are stored on the NMS server computer. This means, any changes you make are stored at a central place and are automatically available to other users. If you wand to add your own icons or images, contact your system administrator to store these files on the server computer.
- A quick start for a new main screen is the tool bar button. It creates a new device icon for each device defined in the VLC/M&C system.

3.16.2.3 The Element Properties Window

The Screen Object Properties window is used to set the properties of a display object within the Screen Editor . Depending on the type of screen element, the window may contain different fields / information, below the element properties window for the common parameter element is shown as an example:



Although the contents of the element properties window depends on the type of element shown, the first line of the window is always the same:

- x --- X location of the element in the window.
- y --- Y location of the element in the window.
- width --- Width of the element's bounding box.
- height --- Height of the element's bounding box.
- element type --- The object type of the element.

Remarks

- If you change an element's position by dragging it with the mouse, this will overwrite any position / size values values edited in the element properties window. Clicking "Apply" before the element is moved saves the edited values.
- Changing the element type also changes the shape of the element properties window. As

the element properties internally are stored in a way compatible to older software versions, the content of some fields may change in a way you might not expect.
- The screen editor uses the same Element Properties window for all elements. If you click to another element in the editor window, the properties of this element get loaded into the properties window.

3.16.2.4 Display Element Types

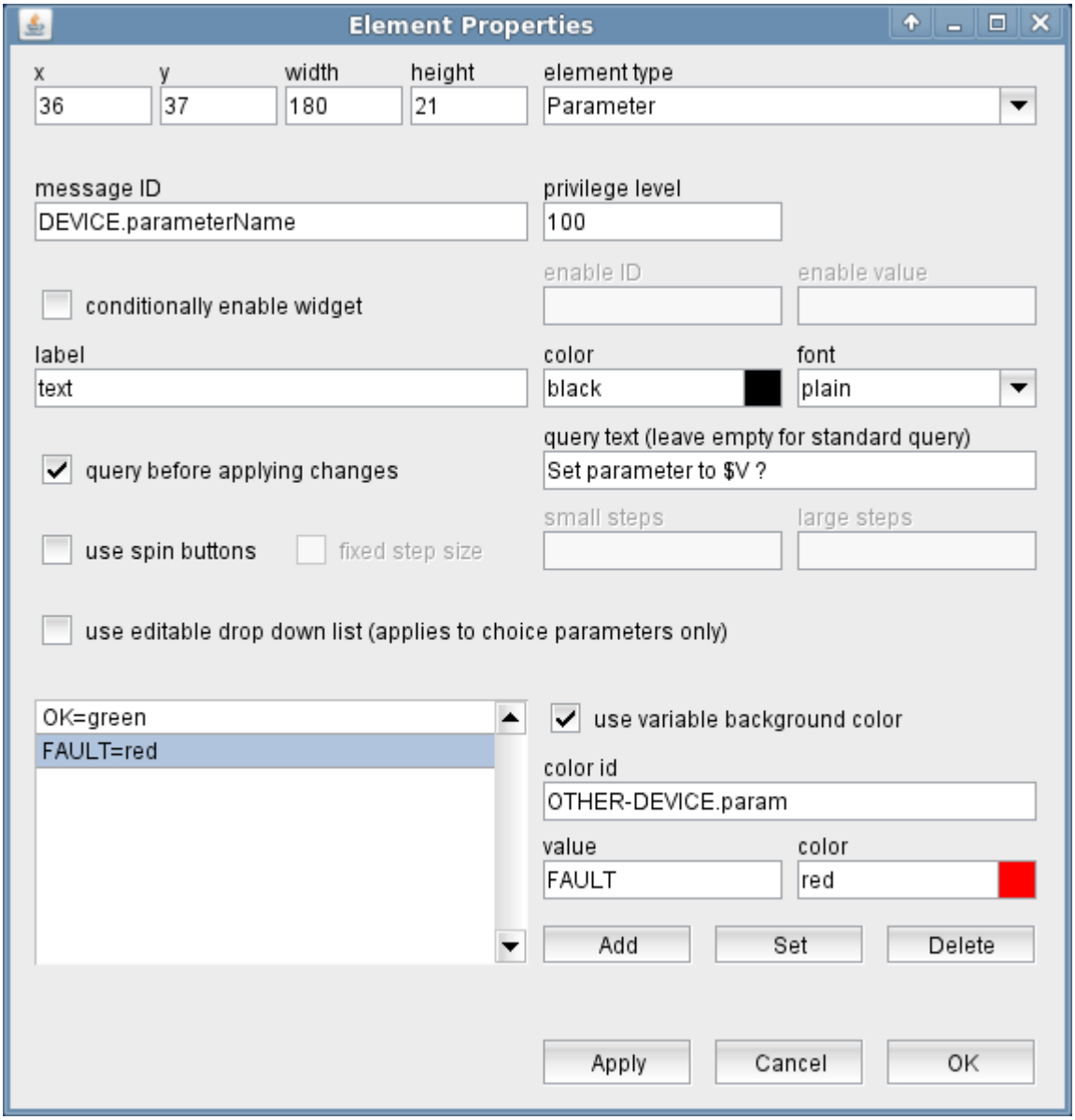
The table below shows the types of display elements the screen editor supports. A detailed description of the individual element types is given on the following pages.

Element Type	***Description***
Parameter	A parameter entry field.
Read-Only Parameter	A parameter display field which does not allow the parameter to be edited.
Radio Buttons	A parameter entry field specially for CHOICE parameters.
Fault	A special display element to display a fault flag. It shows the textual description of the fault and it's state in an entry field frame. The right mouse button shows a context menu to change the fault's priority.
Label	Displays a single line of text.
3D Frame	Draws a sunken 3D frame, may be used to group parameters.
Rectangle	Draws a rectangle.
Line / Arrow	Draws a horizontal/vertical line or arrow.
Icon	Places an icon (GIF or JPG image) into the screen. Optionally the icon can be programmed to change with a parameter value.
EDIT Button	Places an EDIT button into the screen, implicitly forcing the EDIT/TAKE operating method for this screen instead of changing each parameter independently.
TAKE Button	Places a TAKE button into the screen.
LOCK Button	A button to control the lock state for exclusive operation for a number of devices.
Device Icon	Places an icon into the screen which represents a device. This icon will display the device's operating/fault state by it's color / shape.

Element Type	***Description***
Switch Icon	Like the device icon. but additionally displays the actual position of a switch (Meant to be used for block diagrams showing the true signal path.
Frame Button	Defines a button which launches another screen, e.g. for detail views.
Parameter Button	A button which sends a certain parameter value when pressed.
Latching Button	A button which displays/controls a 2-state parameter using it's pressed state.
Gauge	A gauge element, displaying a numeric parameter as a horizontal bar.
Strip Chart	A strip chart element, displaying a numeric parameter as an y/t diagram that automatically advances with 1 pixel / second.
Spectrum Display	The spectrum display element integrates the spectrum display of a spectrum analyzer device in a user screen.
MCP Display	The MCP display element integrates the measurement display of a spectrum analyzer device in multi channel measurement mode in a user screen.
ODM Target List	The ODM Target List element shows the list of targets of aSatService-ACU-ODM antenna controller device. It permits to recall, save or delete target definitions of this type of antenna controller. It is specialized to this antenna controller, does not support other types.
XY Chart	This element shows the relation of two numeric variables in an X/Y diagram, featuring a trace which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.
JSON List	This is a special screen element required for some devices which distribute a state table/list in JSON format.
Thumbnail Icon	A screen element to show thumbnail images from the video processed by an encoder / decoder / gateway device which supports this feature.

3.16.2.4.1 Parameter

The *Parameter* display element is the common component to display and edit most types of M&C parameters. Depending on the data type of the parameter (the data type is detected automatically) the parameter elements appears as textual / numeric entry field, choice box or as display field for read only parameters. In the screen editor, the parameter element is shown as a sunken rectangle.



The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses the parameter the element shall display or edit. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the ' Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- privilege level --- The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.

- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the entry field.
- label --- The label text is drawn above the element.
- color --- The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
- query before applying changes --- Check this mark to make the element show a query before a parameter gets actually commanded.
- query text --- The query text to be shown in the pop-up window. you may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set.
- use spin buttons --- Checking this mark enables "spin buttons" for numeric entry fields. Clicking these small arrow symbols at the right edge of the entry field increases/decreases the value by a certain amount. clicking them with the right mouse button increases/decreases by a larger amount.Enabling spin buttons with automatic step size automatically enables the "use editable drop down list" feature.
- fixed step size --- Checking this mark permits to set user defined increment values for the spin buttons. If not checked, the spin button increment is derived from the least significant digit shown in the field, the large (right mouse button) increment is ten time that.
- small steps --- The spin button increment value to be applied with left mouse button clicks.
- large steps --- The right mouse button spin button increment. The program assumes 10 times the small steps if the large steps field is left empty.
- use editable drop down list / force single click select --- Checking this mark changes the type of drop down list to be used with a choice parameter.Enumerated values normally are shown as a drop down list which pops up if you click into the field. This element type is not keyboard sensitive, you always have to select the value with the mouse which may be inconvenient if the list contains a large number of values.With the editable drop down list option checked you can type into the field and write some characters of the new value you want to set. The element pops up a list of applicable setting containing the typed characters. Clicking to the selected value in the list completes the selection, sets the value.If the widget is configured for a height of 40 pixels or more, it appears as a scrollable list rather than as a drop down box. Checking this mark forces the widget to select a value with a single mouse click, normally a double click is required to select a value.Selecting this option automatically enables "use spin buttons".
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable

addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.

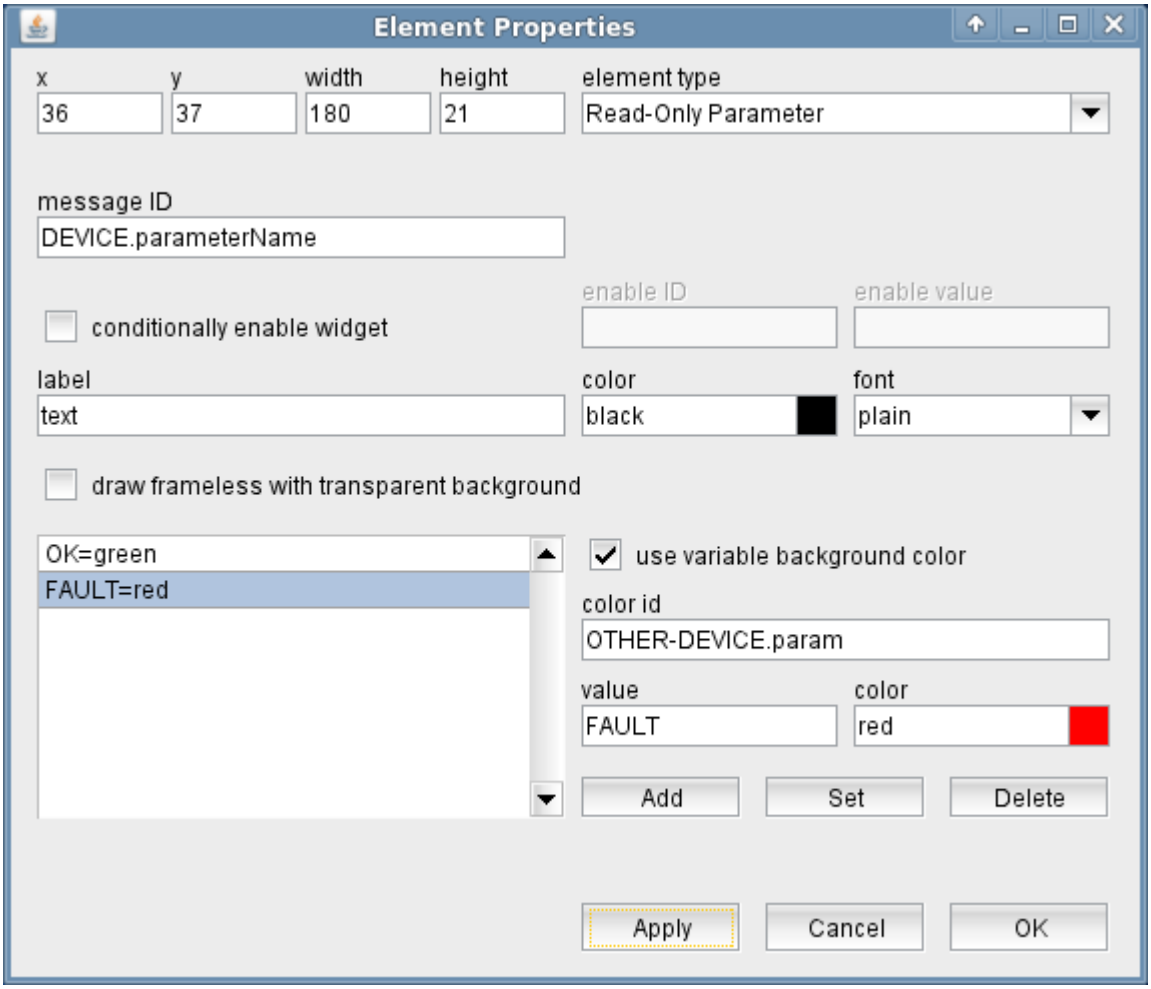
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
  - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
  - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
  - To delete a value / color pair from the list, select the table entry and then click "Delete".

Remarks

- The options "use spin buttons" (automatic step size) and "use editable drop down list" internally use the same use the same property of the screen element. Hence, these options are linked in way that if you enable one of these options, the other one is selected as well. This is done to maintain compatibility to older versions of the software. This linkage means no limitation, as a parameter element never can be a numeric entry field and a choice list at the same time.
- The predefined screens of the software always use the "plain" font and a height of 21 pixels for parameter fields.
- If you enlarge the height for a choice parameter widget to more than 40 pixels, the software uses a list widget with scroll bars rather than a drop down selection box.
- If you enlarge the height for a entry field to more than 50 pixels, the software uses a multi line entry field widget. Multi line entry fields are useful only for very few types of parameters, they are not recommended for general use.

3.16.2.4.2 Read-Only Parameter

The *Read-Only Parameter* display element is used to display M&C parameters read-only. It looks much alike the Parameter element, but never allows to change the parameter it displays.



The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses the parameter the element shall display or edit. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the ' Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- color --- The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
- draw frameless with transparent background --- Check this mark to let the software draw the element without a entry field frame and with transparent background. With this option the Read-Only-Parameter element can be used as a label which changes it's text with the value of a variable. Please note that you cannot use the 'variable background

color' option (described below) together with the frameless / transparent background option.
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
    - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
    - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
    - To delete a value / color pair from the list, select the table entry and then click "Delete".

Remarks

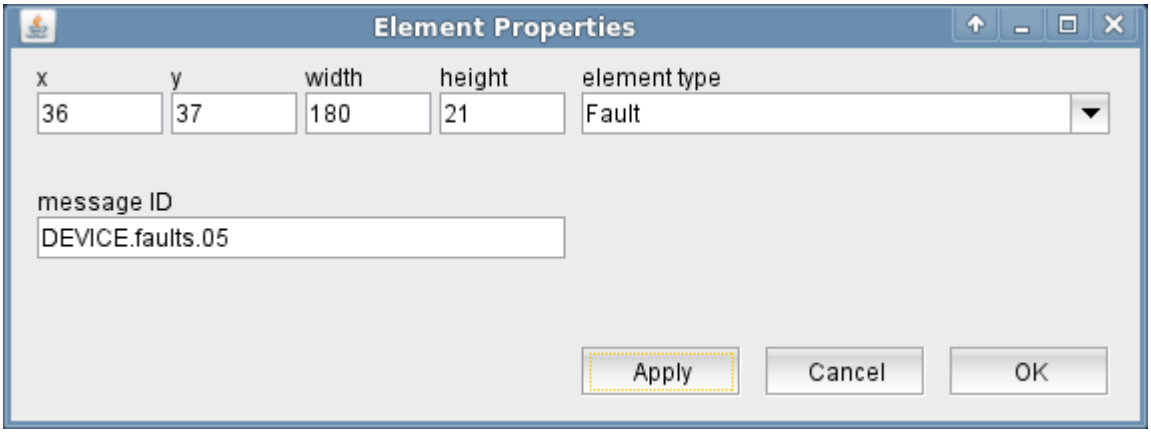
- The predefined screens of the software always use the "plain" font and a height of 21 pixels for parameter fields.

3.16.2.4.3 Fault

The *Fault* display element shows a fault flag. The faults page of the predefined device windows makes use of this display element.

The software treats device faults in a special way. In the variable list of a device driver the fault flags appear as variables called "faults.00" .. "faults.99". The software automatically produces a variable "config.faults.XX" for each fault flag "faults.XX" the driver defines. This configuration variable controls the priority of the fault. The Fault display element now:

- Displays the state if the fault flag (OK/FAULT)
- Displays the fault text (the description for the fault flag is distributed as a message at runtime).
- While the operator holds down the left mouse button on it, the Fault element shows the priority set for this fault.
- With the right mouse button the Fault element shows a context menu to change the fault's priority.

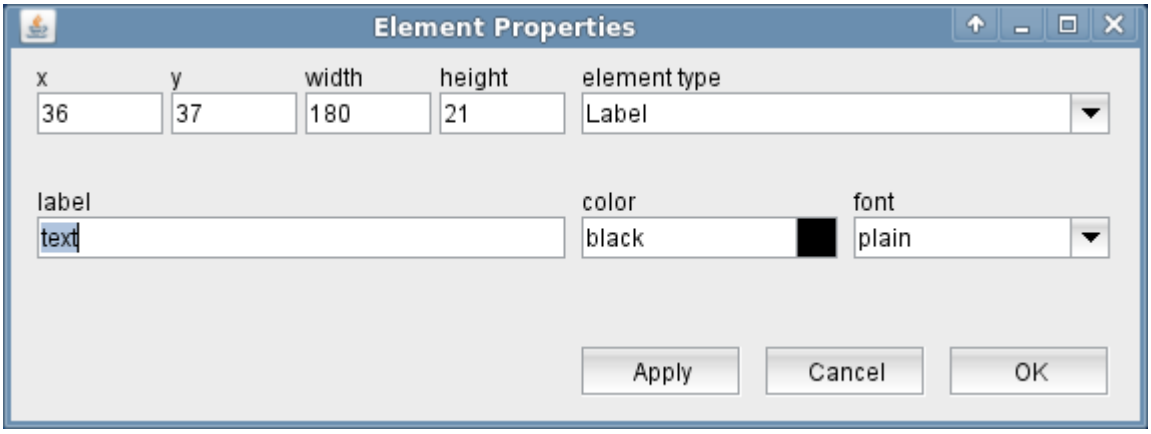


The attributes listed below may be configured to customize a display element of this type:

- Message ID --- The message ID addresses the parameter the element shall display or edit. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.faults.22* for example addresses the variable *faults.22* at a device named *XMIT-1* . With the 'Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.

3.16.2.4.4 Label

The *Label* display element shows a label (a one line text string) with a selectable font / color.

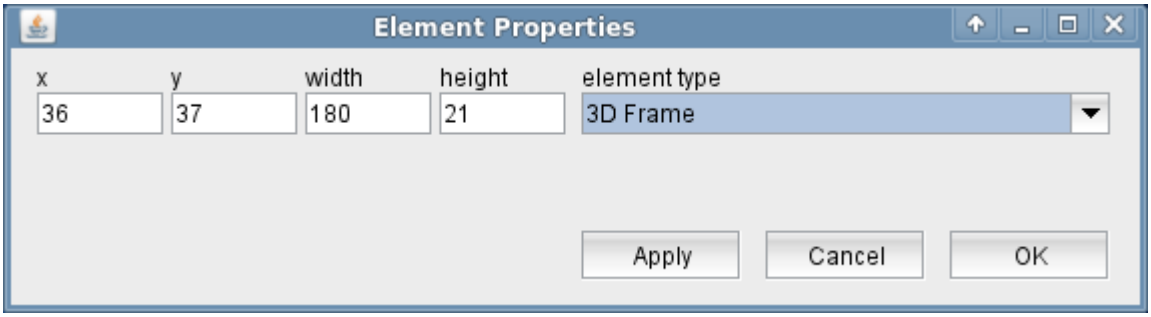


The attributes listed below may be configured to customize a display element of this type:

- text --- The text to be shown.
- color --- The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font to be used for the label.

3.16.2.4.5 3D Frame

The *3D Frame* display element draws a sunken 3D frame, which is intended to be used to group other elements.



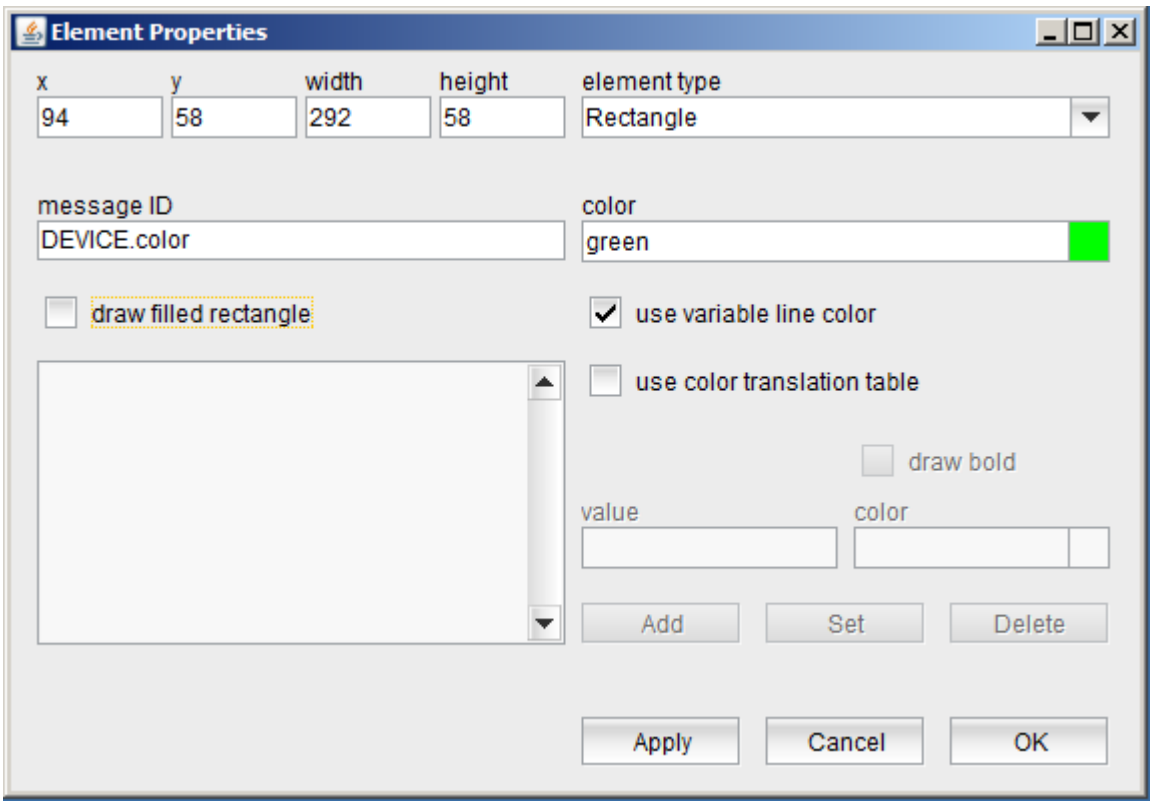
Beside the position in the window there are no user configurable attributes to set.

Remarks

The 3D frame's inside area is transparent, it does not conceal the screen elements it encloses. In the screen editor, however the 3D frame grabs all mouse clicks, so you might not be able to mode the elements which are inside the frame. In this case move the 3D frame to the back by pressing while the frame element is selected. The other elements are now "on top" of the frame element and may be addressed by the mouse independently.

3.16.2.4.6 Rectangle

The *Rectangle* display element draws a rectangle with a selectable color.



The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses a parameter which controls the color of the rectangle. No message ID is required if the rectangle shall be drawn with a fixed color.
- color --- The color to be used to draw the rectangle unless a variable line color is specified. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- draw filled rectangle --- If this mark is checked, a solid rectangle is drawn. The 'draw bold' property in the color translation table has no effect when the rectangle is drawn solid.
- use variable line color --- Check this mark to let the software draw the rectangle with a color specified by the value of the variable specified by 'message ID'. Unless you check the mark 'use color translation table', the software interprest the variable contents as described below in the paragraph 'Color / Line Width Codes'.
- use color translation table --- Checking this mark enables a color translation table which works much like the variable background color feature for entry elements. Depending on the actual value of the variable addressed by 'message ID', the element's line color and thickness is set using the translation table shown in the lower left corner of the dialog. If the actual value of the variable does not match any of the table entries or if no message ID is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- draw bold --- The line thickness specification of the actually selected table entry. If the mark is checked, the line is drawn 3 pixels wide, with the mark unchecked the line width is 1 pixel.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
    - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
    - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
    - To delete a value / color pair from the list, select the table entry and then click "Delete".

Color / Line Width Codes

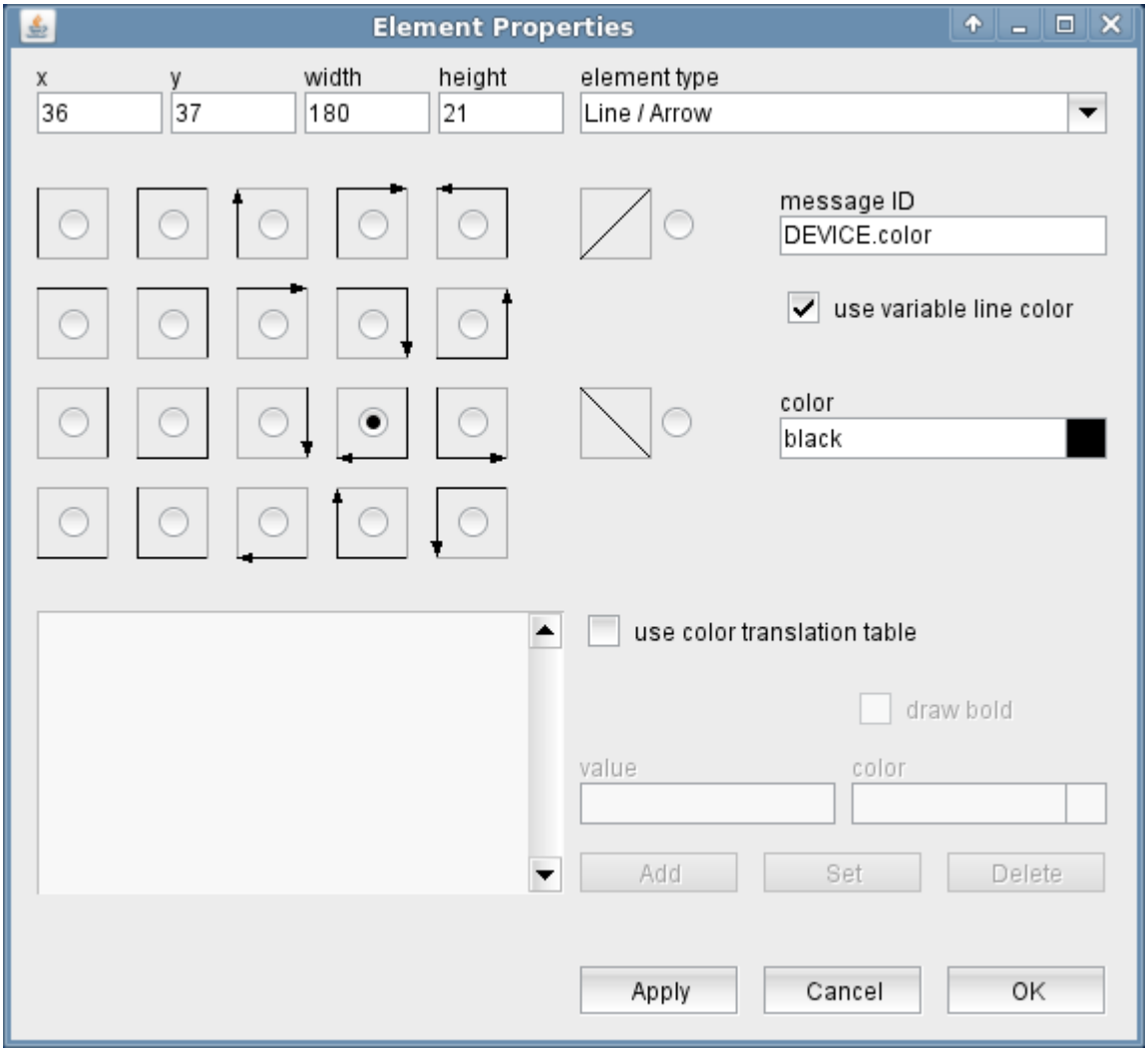
If you check the 'use variable line color' mark without using a color translation table, the software interprets the variable contents as a text value and parses it using the following rules (upper/lower case doesn't matter):

- If the variable value starts with one of the keywords BLACK, WHITE, RED, GREEN, BLUE, YELLOW or GRAY, this color is used.
- If the variable value ends with BOLD, the line is drawn 3 pixels wide (instead of 1 pixel default width).
- If the variable value is TRUE or it contains the text FAULT, the line is drawn red, 3 pixels wide if the variable denotes a fault flag.
- If the variable value is TRUE and the variable does not denote a fault flag, the line is drawn green, 3 pixels wide.
- If the variable value is ON, the line is drawn green, 3 pixels wide.
- If the variable value starts with a '#' character, the following six characters are interpreted as a hexadecimal number, specifying the RGB parts of an arbitrary color. Example: "#FF00FF-BOLD" defines a 3 pixels wide, magenta colored line.
- If none of the rules above matches, the line is drawn with the default color, 1 pixel wide.

The ' Gate-Array ' logical device may be used to combine several state variables in the M&C and generate the color codes expected by the line / arrow element.

3.16.2.4.7 Line / Arrow

The *Line / Arrow* display element draws a horizontal and/or a vertical line and optionally an arrowhead. The line's color is selectable. Strictly speaking the Line / Arrow element is a rectangle with only one or two sides drawn.



The attributes listed below may be configured to customize a display element of this type:

- shape --- To select which parts of the bounding rectangle of the element shall be drawn as a line, select one of the shape symbols.
- message ID --- The message ID addresses a parameter which controls the color of the rectangle. No message ID is required if the rectangle shall be drawn with a fixed color.
- use variable line color --- Check this mark to let the software draw the rectangle with a color specified by the value of the variable specified by 'message ID'. Unless you check the mark 'use color translation table', the software interprest the variable contents as described below in the paragraph 'Color / Line Width Codes'.
- color --- The color to be used to draw the rectangle unless a variable line color is specified. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- use color translation table --- Checking this mark enables a color translation table which works much like the variable background color feature for entry elements. Depending on the actual value of the variable addressed by 'message ID', the element's line color and

thickness is set using the translation table shown in the lower left corner of the dialog. If the actual value of the variable does not match any of the table entries or if no message ID is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.

- draw bold --- The line thickness specification of the actually selected table entry. If the mark is checked, the line is drawn 3 pixels wide, with the mark unchecked the line width is 1 pixel.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
  - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
  - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
  - To delete a value / color pair from the list, select the table entry and then click "Delete".

Color / Line Width Codes

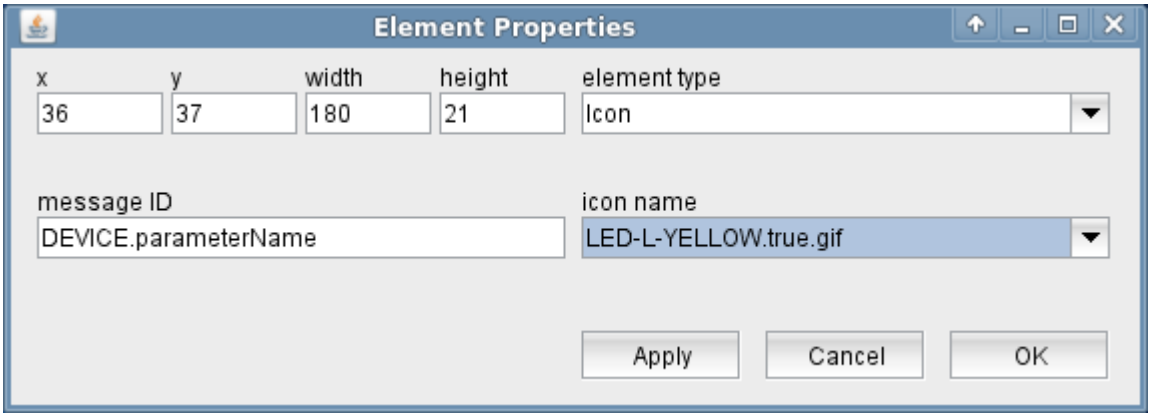
If you check the 'use variable line color' mark without using a color translation table, the software interprets the variable contents as a text value and parses it using the following rules (upper/lower case doesn't matter):

- If the variable value starts with one of the keywords BLACK, WHITE, RED, GREEN, BLUE, YELLOW or GRAY, this color is used.
- If the variable value ends with BOLD, the line is drawn 3 pixels wide (instead of 1 pixel default width).
- If the variable value is TRUE or it contains the text FAULT, the line is drawn red, 3 pixels wide if the variable denotes a fault flag.
- If the variable value is TRUE and the variable does not denote a fault flag, the line is drawn green, 3 pixels wide.
- If the variable value is ON, the line is drawn green, 3 pixels wide.
- If the variable value starts with a '#' character, the following six characters are interpreted as a hexadecimal number, specifying the RGB parts of an arbitrary color. Example: "#FF00FF-BOLD" defines a 3 pixels wide, magenta colored line.
- If none of the rules above matches, the line is drawn with the default color, 1 pixel wide.

The ' Gate-Array ' logical device may be used to combine several state variables in the M&C and generate the color codes expected by the line / arrow element.

3.16.2.4.8 Icon

The *Icon* display element shows an arbitrary GIF/JPEG picture. The image file must reside in the *./images* directory on the M&C/NMS server.



The attributes listed below may be configured to customize a display element of this type:

- message ID --- You can make the image change with a parameter value. The message ID addresses the parameter which selects the image to display (see below for details). The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the ' Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- icon name --- Selects the name of the image to display.

Using an icon to display a parameter

If you specify a message ID with the element properties of an *Icon* , the image displayed will change with the parameter value addressed by the message identifier.

You must supply one image file for each value the parameter may have. The parameter value must appear in the file name just in front of the gif/jpg suffix, enclosed in dots. All images used for one parameter should be of the same size. If the *Icon* element receives a parameter value it does not find an image file for, a black rectangle is displayed instead of an image.

Example: Let's assume you want to visualize a TX ON/OFF switch with an antenna symbol changing it's color with the position of the switch. Supply two images called "transmission.ON.gif" and "transmission.OFF.gif" which symbolize these states as you want. Notice that the ON/OFF keywords in the file names *exactly* reflect the possible values of this parameter and that the file names only differ in this point. Select one of image files for the icon element and enter the message Id of the TX-ON/OFF parameter into the appropriate field.

The mechanism used to select the image file to display mostly limits the usage of dynamically changing icons to CHOICE parameters. Merely numeric (integer) parameters with a very limited number of possible values come into question, too.

Using an icon to display a device summary fault

With device summary faults ("DEVNAME.fault"), and subsystem faults ("SUBSYSTEM. ... .fault")

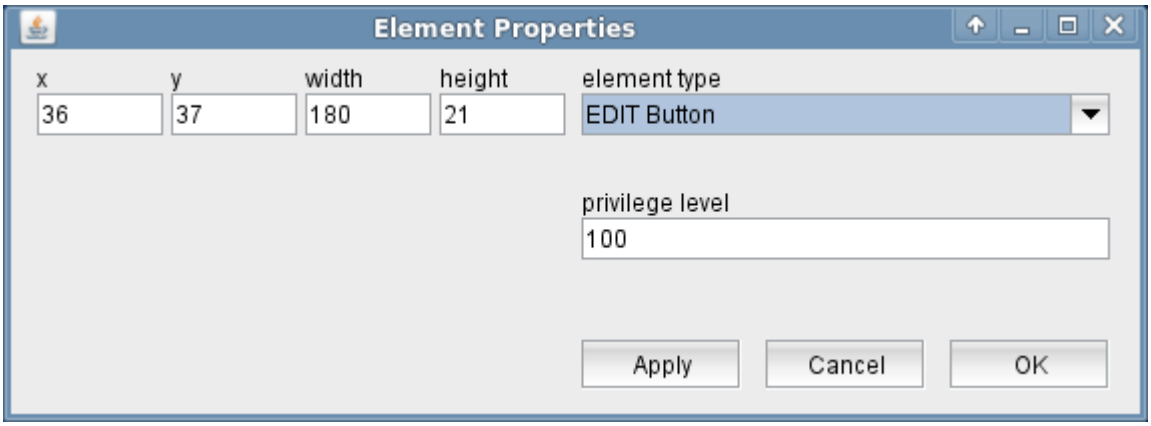
the *Icon* element translates the fault state to "true" for FAULT and "false" for "OK.". This is because the sat-nms software provides the individual fault flags in a device driver as boolean values, the summary fault however is represented by a text which can be one of "OK.", "WARNING" and "Summary FAULT". Translating the OK/FAULT values lets you use the same image files for individual and summary fault states.

To display a device summary fault as an color changing icon, you have to provide three image files, for example:

MyIcon.false.gif for the "OK" state **MyIcon.WARNING.gif** for the "WARNING" state **MyIcon.true.gif** for the "FAULT" state

3.16.2.4.9 EDIT Button

The *EDIT Button* display element places an EDIT button to the screen. It appears as a simple button with the label "EDIT" on it. You define one (and only one) EDIT button in a screen which shall be operated following the EDIT / TAKE operating method explained below. If you have an EDIT button in a screen you must have e TAKE button , too.



The attributes listed below may be configured to customize a display element of this type:

- privilege level --- With this attribute you specify the minimum privilege level an operator must own in order to be permitted to press this button.

The EDIT / TAKE operating method

The M&C software uses the same display elements to show the values read from a device (monitoring) and to change parameter settings on an operator's request (control). The operating situation determines whether a display element contains a monitored value or an edited one.

The software usually changes each parameter as you edit it. More precisely, if you edit a value in an entry field, the field's background becomes yellow to point out that the value is an edited one. If you leave the field or if you press RETURN, the edited value is sent to the device. CHOICE parameters behave slightly different, but in a similar sense.

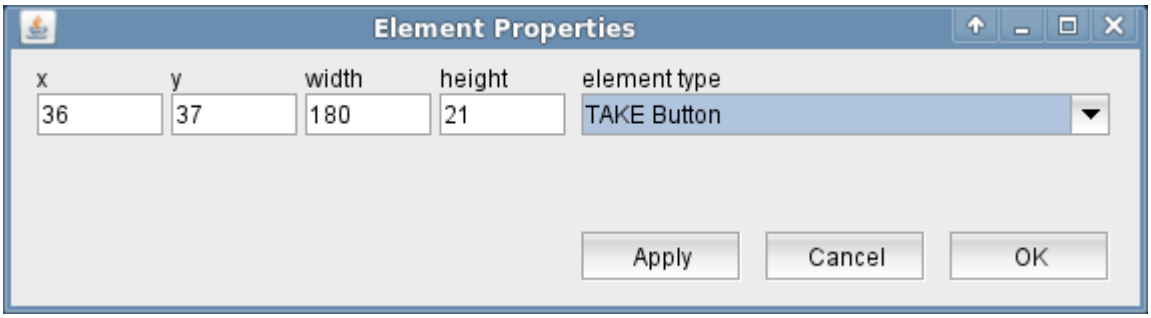
Now, if an EDIT button is defined in the screen, it behaves completely different. All display

elements in the screen show the values monitored at the device in the default state of the screen. You can't change any parameter in this state unless you press the EDIT button. With the EDIT button pressed (it's a latching one) the screen turns into "edit" state and stops updating the screen elements with the values monitored from the device. Now you can change the parameter values, however a changes value will not be sent to the device unless you press the TAKE button. When doing so, all parameter values from this screen are sent to the device at one time. The screen returns to the default state in order to show the values read back from the device.

Using the EDIT / TAKE operating method disables the "intelligent" change parameter ranges, some device drivers support. A modem device driver may dynamically change the valid range of the symbol rate depending of the FEC actually selected. With the EDIT / TAKE operating method, the device driver does not receive the FEC setting as you edit it and therefore cannot change the symbol rate range accordingly. For this reason you should use the EDIT / TAKE operating method only where this is a strong requirement for some reason, not as the default operating style for all screens.

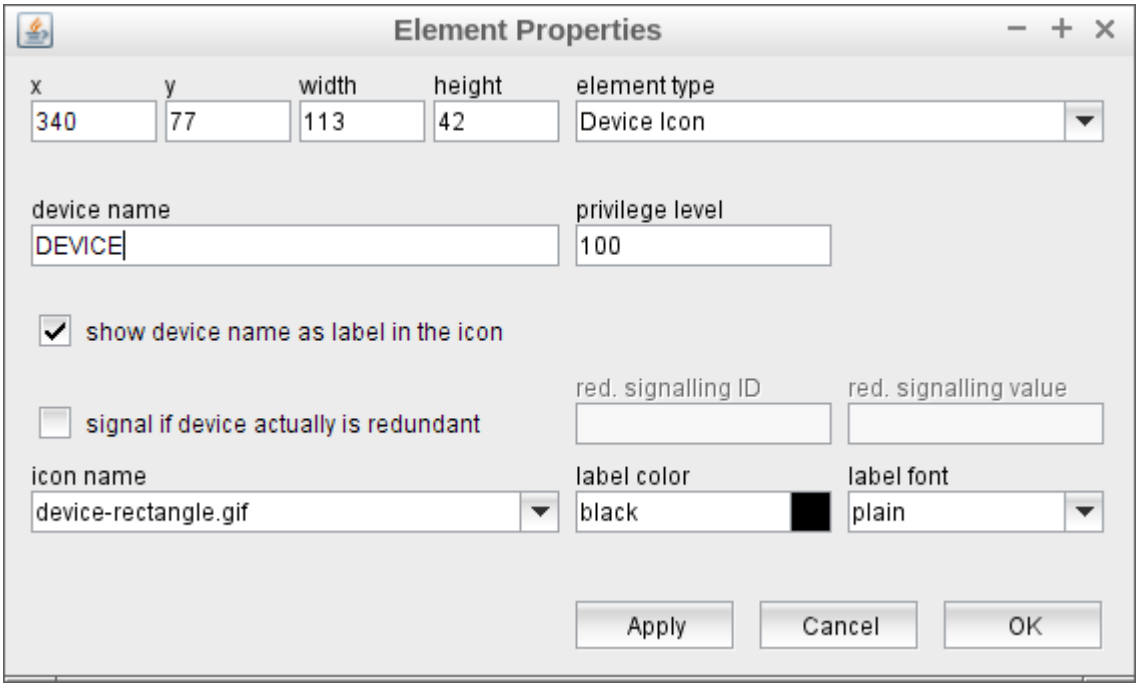
3.16.2.4.10 TAKE Button

The *TAKE Button* display element places an TAKE button to the screen. It appears as a simple button with the label "TAKE" on it. The TAKE button complements the EDIT button described above in this document. It sends the changed parameter values to their destination when pressed. A detailed description of the EDIT / TAKE operating method for M&C screens is given at the EDIT Button chapter.

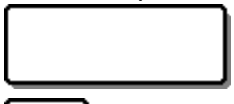


3.16.2.4.11 Device Icon

The *Device Icon* display element represents a device in the M&C user interface. It displays the status of the device by it's color/shape and gives access to the Device Window for this particular device by a double mouse click. The right mouse button launches a context menu with common operations for the device.



The attributes listed below may be configured to customize a display element of this type:

- device name --- The name of the device this element stands for.
- privilege level --- The privilege level necessary to use this device icon. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to use this icon.
- show device name as label in the icon --- Check this mark to show name of the device as a label in the icon. Un checking may be useful when using very small icons where the label would not fit inside or when a label shall be used which differs from the name of the device.
- signal if device actually is redundant --- Check this mark to apply a special coloring for devices which are in the redundant signal path of a redundancy switching facility.
- red. signalling ID --- The message ID of a parameter which gives the information if the device actually operates as the redundant one or if it is operational. For a 1:1 TWTA redundancy this could be the position parameter of the waveguide switch which selects the signal from the two TWTAs
- red.singalling value --- If the value of the parameter addressed by above message ID matches the value given in this field, the device icon is shown as "redundant" (depends in the icon set used, the factory icon sets use a blue background color for this).
- icon name --- Selects the name of the image to display. The sat-nms software at least provides three predefined icons for this purpose:
  -  --- device-rectangle.gif
  - --- device-square.gif
  - --- device-minirect.gif

Your installation probably will contain some more device icons to select.

- label color --- The color attribute of a parameter element sets the color of the device name shown with the icon.
- label font --- The font attribute applies to the device name, too.

Using customized images

You may supply your own images for the device icon display element. To do so, place seven images into the *./images* directory named "xxxxx.gif", "xxxxx-F.gif" (fault), "xxxxx-C.gif" (communication fault), "xxxxx-S.gif (fault suppressed)", "xxxxx-R.gif" (redundant), "xxxxx-O.gif" (out of service) and "xxxxx-M.gif" (maintenance). You should consider the following:

- All images should be of the same size.
- The software places the device name at the lower left corner of the image using the selected font and color. All four images should use appropriate colors in this area to keep the device name visible.
- Beside GIF-images the software also accepts PNG images (file extension .png) and JPG images (file extension .jpg). All images of a set must be of the same type (GIF/PNG/JPG).

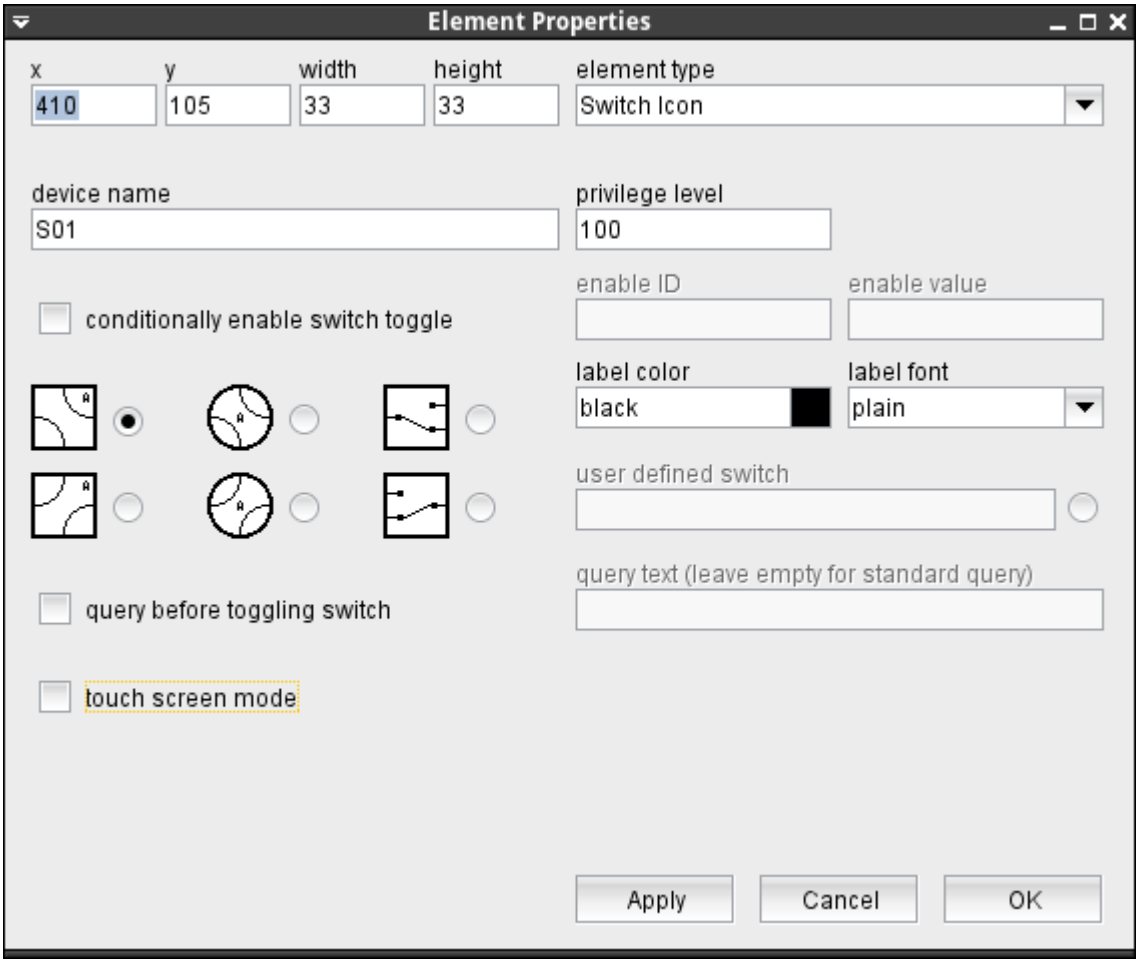
Using a customized 'transmitting' emblem

For devices which provide the 'info.signal.on' parameter to report that the device is actually transmitting or receiving, the device icon shows an emblem at it's upper right corner. The software uses the 'signal-on-emblem.png' image as a default for this.

You may define individual emblems for each image set by placing a "xxxxx-X.gif" (or .png/.jpg) image in the ./images folder with xxxxx being the base name of the device icon. The software will use this file for the transmitting emblem of this particular icon set. The file extension of the emblem must match the extension of the base icon image.

3.16.2.4.12 Switch Icon

The *Switch Icon* display element is a special version of the Device Icon element which may be used to visualize the position of a switch in a user interface screen designed as a block diagram. The switch icon has all capabilities of a plain device icon display element. The context menu shown with this right mouse button additionally contains an option to toggle the switch position. The switch icon element -- unlike the plain device icon element -- does *not* support user defined images.



The attributes listed below may be configured to customize a display element of this type:

- device name --- The name of the device this element stands for. You also may enter the complete ID of the parameter which controls the switch position for devices which manage more than one switch or use a non-standard switch position parameter name.
- privilege level --- The privilege level necessary to toggle the switch position. To change the device mode (OPERATIONAL, OUT-OF-SERVICE), at least a privilege of 100 is necessary, or the value set here if this is higher.
- conditionally enable switch toggle --- The toggle function of a switch icon may be locked unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.
- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" must match this value in order to enable the toggle option of the switch icon.
- shape --- Select the shape of the icon from the given switch symbols or a user defined switch. .ts user defined switchThe (base-) name of a set of userdefined switch icons. See below how to use your own switch icons.
- label color --- The color attribute of a parameter element sets the color of the device name shown with the icon.

- label font --- The font attribute applies to the device name, too.
- query before toggling switch --- Check this mark to make the element show a query before a switch toggle gets executed.
- query text --- The query text to be shown in the pop-up window. you may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set.
- touch screen mode --- Selecting this option makes the switch icon behave differently, optimized for touch screen usage. You may toggle the switch position by touching the icon twice. This first time you touch it, the icon shows a turquoise ball in the upper left corner. This ball disappears after 2 seconds. If you touch the icon a second time before the turquoise ball disappears, the switch is actuated.There is no right mouse button context menu in this mode, no way to go into the device screen or set the device OUT-OF-SERVICE.

Remarks

Configuring a confirmation question as described above, only affects the 'Toggle position' function provided by the switch icon's right mouse context menu. Setting the switch position from the device dialog always works without a confirmation request.

User defined switch icons

The sat-nms software comes with 6 predefined switch icons. You may add your own icons if none of the predefined switches matches your needs.

For a user defined switch you have to create a set of ten PNG image files / icons. Each icon shows one combination of switch position and device state. All icons of a set must be of the same size, the files must be located in the *images* subdirectory of the sat-nms server installation and they must follow the naming scheme described below.

The file name of a switch icon consists of the user defined name, follwed by a dash and a 2-character status code. All switch icons must be PNG images and end with the file suffix ".png" (lower case). The first character of the status code reflects the switch position, the second one the device's fault/operation state. The following codes are defined:

pos.	***character***	***description***
1	A	switch is in position A (or OFF)
1	B	switch is in position B (or ON)
2	N	switch device is in normal operation, no fault is pending
2	W	switch device shows a warning
2	F	switch device shows a fault

pos.	*character*	*description*
2	S	switch device is set FAULT-SUPPRESSED
2	O	switch device is set OUT-OF-SERVICE

Combining the two switch positions with five device status codes results in totally 10 icons which show all possible states of a switch. If you create a switch named "myswitch", the files would be:

myswitch-AN.png myswitch-AW.png myswitch-AF.png myswitch-AS.png myswitch-AO.png myswitch-BN.png myswitch-BW.png myswitch-BF.png myswitch-BS.png myswitch-BO.png

Replacing the predefined icons

In in your sat-nms installation mostly user defined switch symbols are used, you may replace some or all of the predefined switch symbols offered by the Switch Icon dialog with your favorite icons. To configure this, edit the file "client.properties" in the base directory of the sat-nms server installation and add some or all lines of the following:

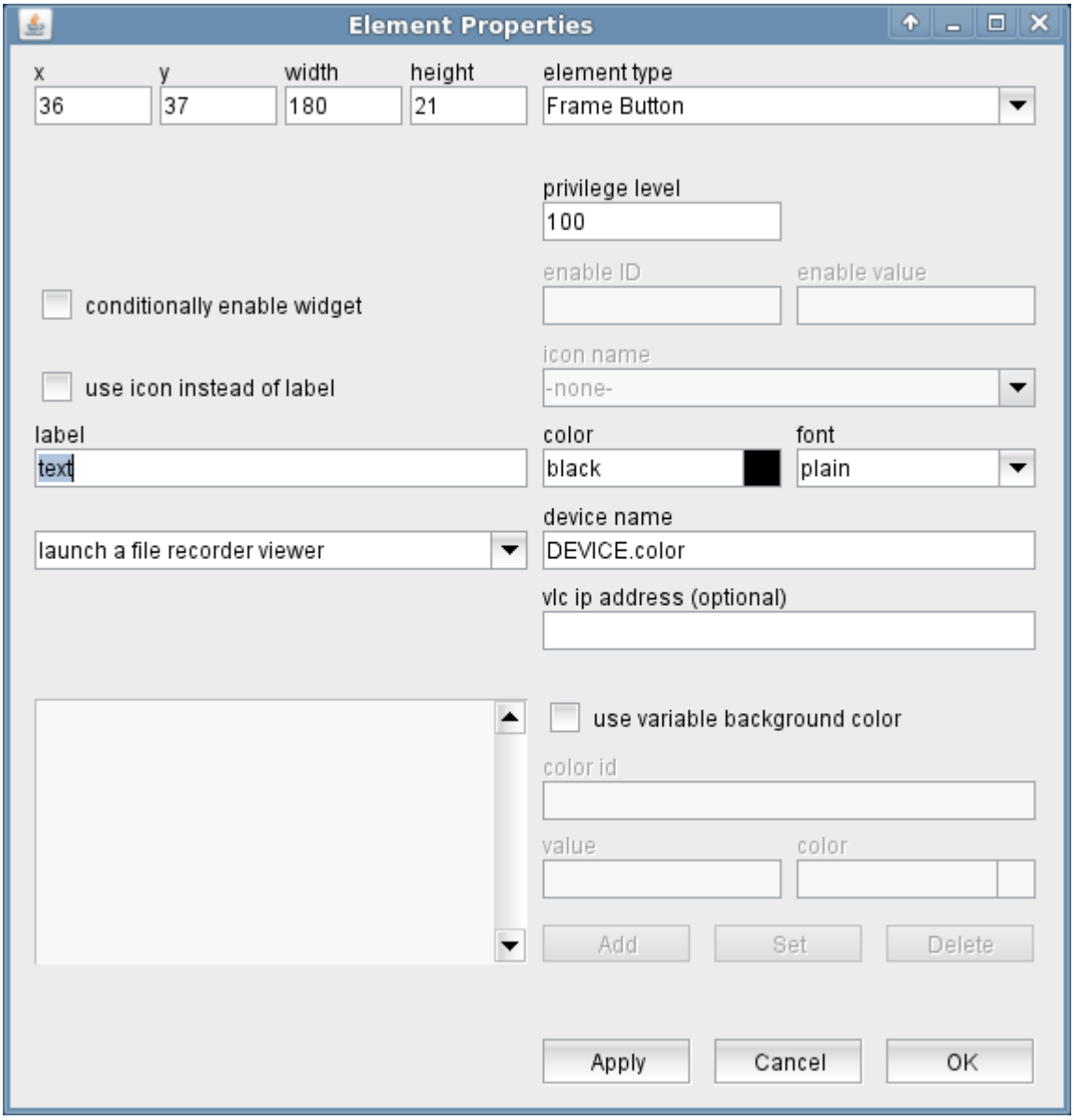
gui.editor.predefinedSwitch.1=myswitch1 (replace myswitch1 by the name of the icon set to use) **gui.editor.predefinedSwitch.2=myswitch2 gui.editor.predefinedSwitch.3=myswitch3 gui.editor.predefinedSwitch.4=myswitch4 gui.editor.predefinedSwitch.5=myswitch5 gui.editor.predefinedSwitch.6=myswitch6**

Please note, this does not replace / delete the factory icons. Screens which make use of the factory switches will still use them as before and you are still able to edit screens with factory switch icons by explicitly define their names in the *user defined switch* field. The names of the factory switch icons are *wgswitch* , *wgswitch-mirrored* , *cxswitch* , *cxswitch-mirrored* , *lswitch* and *rswitch* .

User defined switch icons may be of any size, even larger than the factory icons. When used as predefined switch icons, such large icons will show up in the dialog with their upper left corner only. In the screen the icons gets displayed with it's full size.

3.16.2.4.13 Frame Button

The *Frame Button* display element creates a button which launches another user interface screen if pressed. The screen to be launched may be another user defined screen or a predefined one.



The attributes listed below may be configured to customize a display element of this type:

- privilege level --- The privilege level necessary to activate this button. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- FRame buttons may be locked to disabled state unless another parameter matches a given value. Check this mark to activate this feature.
- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the entry field.
- use icon instead or label --- Check this mark in order to make the button show an icon instead of a label text.
- icon name --- Provides a list of all available icons / images on the sat-nms server.
- font --- The font to be used for the button label.

- label --- The label text is drawn on the button.
- color --- The color of the button label. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font of the button label.
- button function --- with this field you select the function of the button. The available functions are listed below. The meaning of the parameters right beside the function selector changes with the function selected.
- *launch a child screen* --- Launches a new screen in a separate window. Parameter 1: screen name Parameter 2: n/a
- *replacescreenwithanotherone* --- Replaces the actual screen by another one, re-uses the same window. Parameter 1: screen name Parameter 2: n/a
- *launch a tty setup window* --- Launches a dialog to edit the line settings of a serial interface. Parameter 1: interface name, e.g. ttyS2 Parameter 2: n/a
- *launch a device setup window* --- Launches a dialog to edit the setup parameters of a given device. Parameter 1: name of the device Parameter 2: n/a
- *launch a load preset dialog* --- Launches a dialog to select and apply a preset to a given device. Parameter 1: must contain the name of the device, followed by a comma and the name of the device driver used by the device. Example: MODEM-1,Comstream-CM601A-SCPC Parameter 2: if this parameter is not empty, the preset load dialog initializes it's search pattern field with this value. Only presets containing the pattern string (character case doesn't matter) will appear in the list.
- *launch a text viewer window* --- Launches a multiline text view window which shows the actual contents of the given parameter. Parameter 1: The message ID of the parameter to show. Parameter 2: n/a
- *launch a tracking viewer* --- Launches a tracking report window for the Antenna-Tracking logical device. The tracking report window shows azimuth or elevation together with the beacon level over the recent 48 hours. Parameter 1: The message ID to show. Typically this is the "state.mode4" Parameter of an Antenna-Tracking or a SatService-ACU-ODM device. Parameter 2: n/a
- *launch a az/el viewer* --- This is much like the tracking viewer, but the window plots elevation versus azimuth in a quadratic diagram. Parameter 1: The message ID to show. Typically this is the "state.mode4" Parameter of an Antenna-Tracking or a SatService-ACU-ODM device. Parameter 2: n/a
- *launch a file recorder viewer* --- Launches a graphical presentation screen for data recorded with a File-Recorder logical device. Parameter 1: Name of the File-Recorder device and (optionally) a preset number 1..8 to be invoked after the data has been loaded. A ">" may be appended to the preset number in order to make the diagram show recent data instead of the data shown at the time when the preset was stored. Example: "FILEREC-12>" shows the dat a of the FILEREC-1 device, invokes the preset number 2 and shows the data recently recorded by the file recorder device. Parameter 2: In an NMS environment you must supply the VLC's IP address in this field. The VLC must be reachable directly and permanently from the client PC to make this feature work safely. In a M&C system leave the field empty
- *launch a spectrum viewer* --- Launches the device window of a spectrum analyzer and optionally sets an arbitrary number of parameters. Parameter 1: The name of the

spectrum analyzer device Parameter 2: An optional macro list. This may contain zero or more parameter ID - value pairs. These settings may be used to preset the spectrum analyzer certain settings but also to control switches or other devices in the signal path. Example: S03=B, SA1.center=12345.666 If the operator clicks to this button, the switch S03 is commanded to position B, the spectrum analyzer SA1 is set for a center frequency of 12345.666 MHz and the spectrum analyzer window gets opened.

- *launch an m&c client* --- Launches another M&C client program which connects to the given IP address. The M&C client runs in a separate JVM. It tries to login at the target machine with the same name and password the parent program uses. Parameter 1: The address to connect to Parameter 2: n/a
- *launch another java runtime* --- Runs an arbitrary Java program by starting another JRE process. Note, that all command line arguments including the classpath specification must be provided. These arguments probably will be platform specific, hence a user interface depending on this feature only will work on the platform it is configured for. Parameter 1: The command line parameter for the java program Parameter 2: n/a
- *launch a web browser window* --- Launches the platform's standard web browser and shows the given URL. Parameter 1: The URL to show. To show a certain page of the online manual, simply use the 4 digit topic number of the page as the URL parameter. To show an extern web page, e.g. the web based user interface of a sat-nms ACU ODM, enter a URL starting with "http://...". Example: " **http://192.168.2.81** ". Parameter 1 must be a valid URL string in this case, space characters must be escaped as "%20". The BROWSER button this way also can be used to operate a device which provides a web user interface. In device-oriented screens, the character '@' gets expanded to the device's IP address if contained in the URL string. Example: " **http:///** " lets the browser call the top page of the device's web user interface. Parameter 2: n/a
- *start a program on the client pc* --- Starts a program on the client computer. The program must be specified with the full path to the executable file. Parameters for the program may be added, separated space characters. If the path name itself contains space characters, in must be enclosed in double quotes.
- *launch a VideoLAN media player window* --- Starts the VLC media player in a new window to show live video from a certain device. This works with devices like webcams, if the device driver is prepared for this. See the paragraph "VideoLAN media player" below for more details. Parameter 1: The name of the device for which the VLC player shall show the video stream. When used in a device window, parameter 1 is ignored and the button uses the name of the device of this window. Parameter 2: n/a
- *launch a Pointing-Robot import screen* --- Launches a file import window which lets you import pointing table, TLE or Intelsat ephemeris data files from the client PC to the M&C server compunter into the directory where the Pointing-Robot expects these files. Parameter 1: n/a Parameter 2: n/a
- *navigate to a given treeview node* --- Makes the button a navigation button for the treeview application. Clicking the button changes the panel at the right side of the tree view window as if the tree node described by the path specification would have been clicked. Parameter 1: The tree node (path) specification for the button. The path consists of the display names of the tree nodes making up the path to the desired node, separated by '.' characters. If the button for example shall show the page for a subsystem called "Stream-1" contained in a subsystem called "Streams", then the path specification for the button is **Streams.Steam-1** Parameter 2: n/a

- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background. The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
  - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
  - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
  - To delete a value / color pair from the list, select the table entry and then click "Delete".

VideoLAN media player

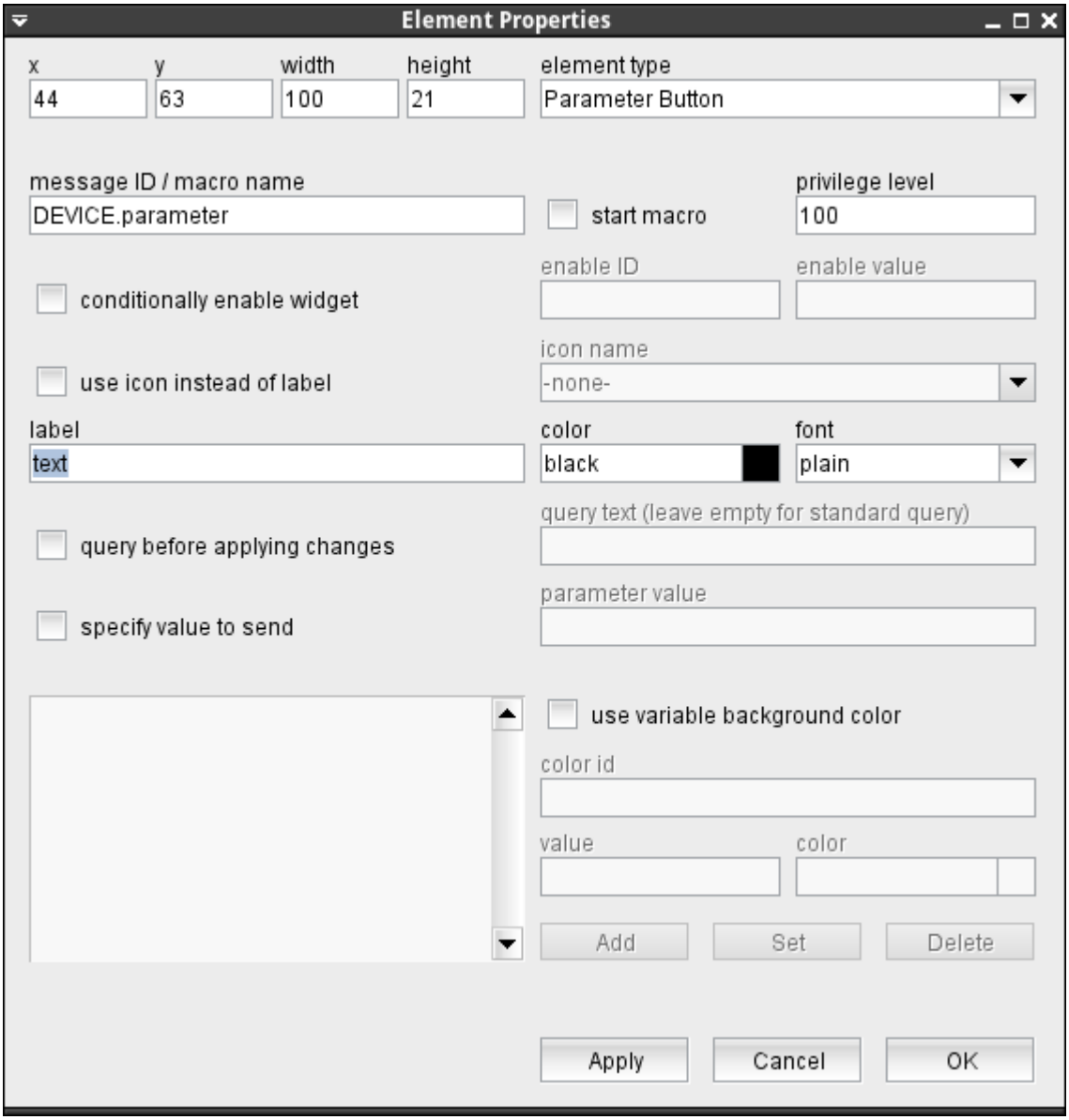
The Frame Button may be configured to launch a VideoLAN media player (VLC) in order to show a live video stream sent by any device capable for this. To use this feature, the VideoLAN media player must be installed on the client machine and the sat-nms driver of the device sending the video stream must be prepared for this.

The frame button expects the device driver to provide a variable called *videoLAN.cmds* which must contain all command line parameters the VLC media player requires to show the desired stream. In the simplest case this is an URL specifying the video stream, but other command line parameters for the VLC player may be used as well.

To launch the VLC media player on Microsoft Windows clients, the program must know the complete path to the executable "vlc.exe". In the M&C server's configuration file 'client.properties' you may set this path. If required, multiple variants may defined, one for each client PC. You find more details on this in the factory default 'client.properties' file.

3.16.2.4.14 Parameter Button

The *Parameter Button* display element is a button which sends a parameter value if pressed. A frequently used application for the parameter button is a RF-OFF button which sends a "tx.on=OFF" to a certain device. Beside this, a parameter button also may be programmed to play a parameter setting macro.



The attributes listed below may be configured to customize a display element of this type:

- message ID / macro name --- The message ID addresses the parameter the button shall control. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable*tx.on* at a device named *XMIT-1* . With the '[Device Driver Index](#) ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.If the "start macro" mark ist checked, this button acts as a macro player button instead of sending a parameter. In this case the field contains the name of the macro to play. If used in an NMS, the macro with this name must be present in the "./macros" subdirectory of the NMS installation.
- start macro --- Selects if the button shall start a macro (checked) or send a certain parameter (unchecked).
- privilege level --- The privilege level necessary to access this button. The predefined screens of the software use 100. In your own screens you may set certain parameters to

higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- Parameter buttons may be locked to disabled state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.
- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the button.
- use icon instead or label --- Check this mark in order to make the button show an icon instead of a label text.
- icon name --- Provides a list of all available icons / images on the sat-nms server.
- label --- The label text is drawn on the button.
- color --- The color of the button label. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font to be used for the button label.
- query before applying changes --- Check this mark to make the element show a query before a parameter gets actually commanded.
- query text --- The query text to be shown in the pop-up window. you may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set.
- specify values to send --- Checking this mark let you define the parameter value to be sent with a button click. Without this option the button sends the label text as a parameter value.
- parameter value --- The value to be sent when the button is clicked.
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
    - To add a new value / color pair to the list, fill the value and color fields, then click to

"Add".
- To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
- To delete a value / color pair from the list, select the table entry and then click "Delete".

Using the Parameter Button to control device presets

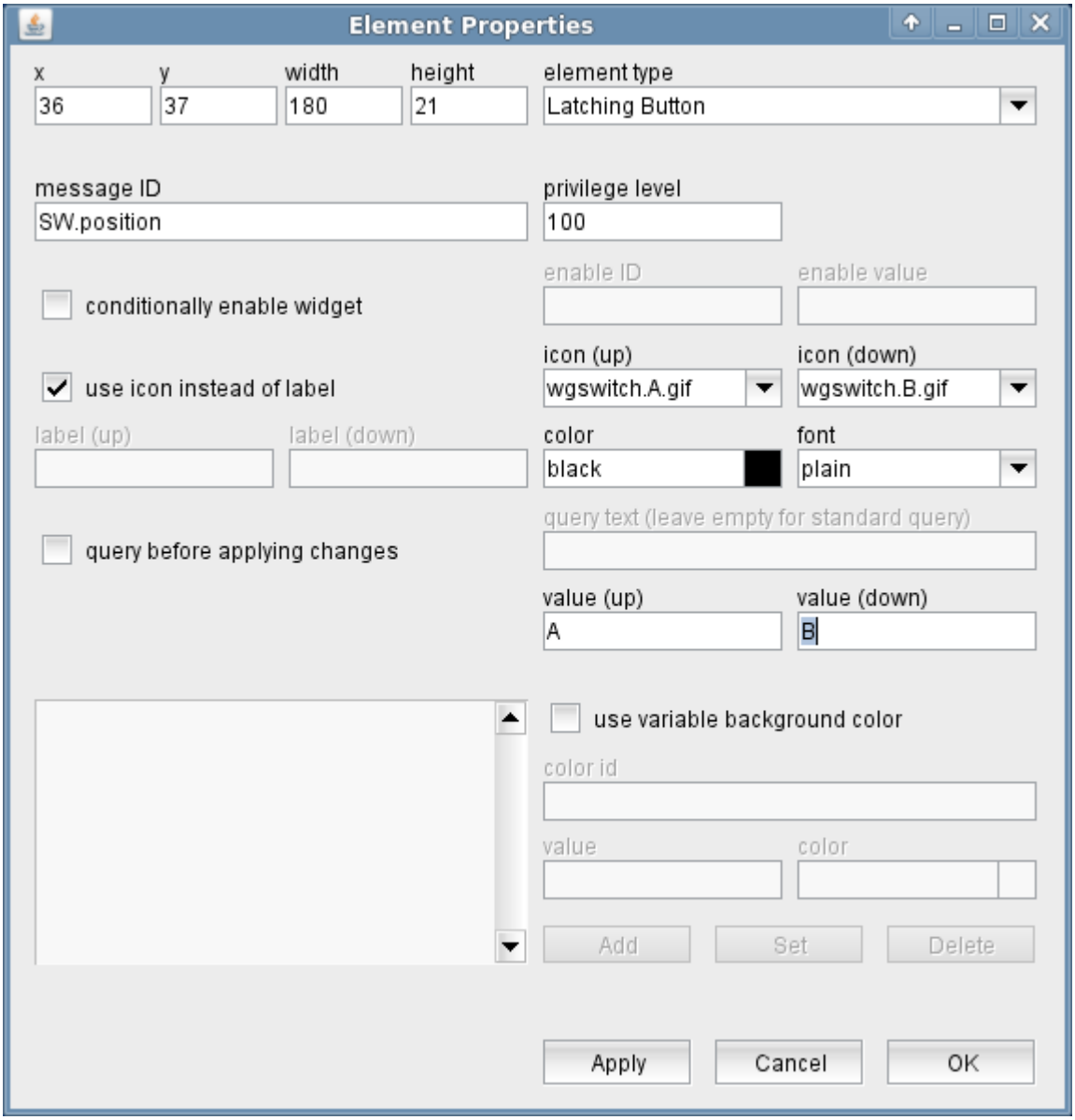
Each device in the sat-nms software knows two hidden variables which control storing and applying of device presets. The variables are called 'loadPreset' and 'savePreset'. Sending a string value to one of these variables will load or save a device preset with the name given as the variable value. Please note, that this feature uses preset files stored on the M&C/VLC itself. Hence, in a NMS environment these presets are only available for the particular VLC and these presets cannot be managed by the standard preset load/save functions in the device window toolbar in this case.

3.16.2.4.15 Latching Button

The *Latching Button* display element works much like the Parameter Button element described in the previous chapter, but specialized to show and control an enumeration parameter which knows exactly two states (e.g. on/off or true/false).

When the operator changes the state of the button by clicking it once, the latching button sends the 'other' parameter value to the device. On the other hand, if some other instance in the system changes the parameter state, the latching button recognizes this and changes the up/down state of the button accordingly.

Like the parameter button, the latching button may be labeled with text or an image. To reflect the actual state, the latching button always is configred with two text string or two image names which are shown according to the actual parameter value.



The attributes listed below may be configured to customize a display element of this type:

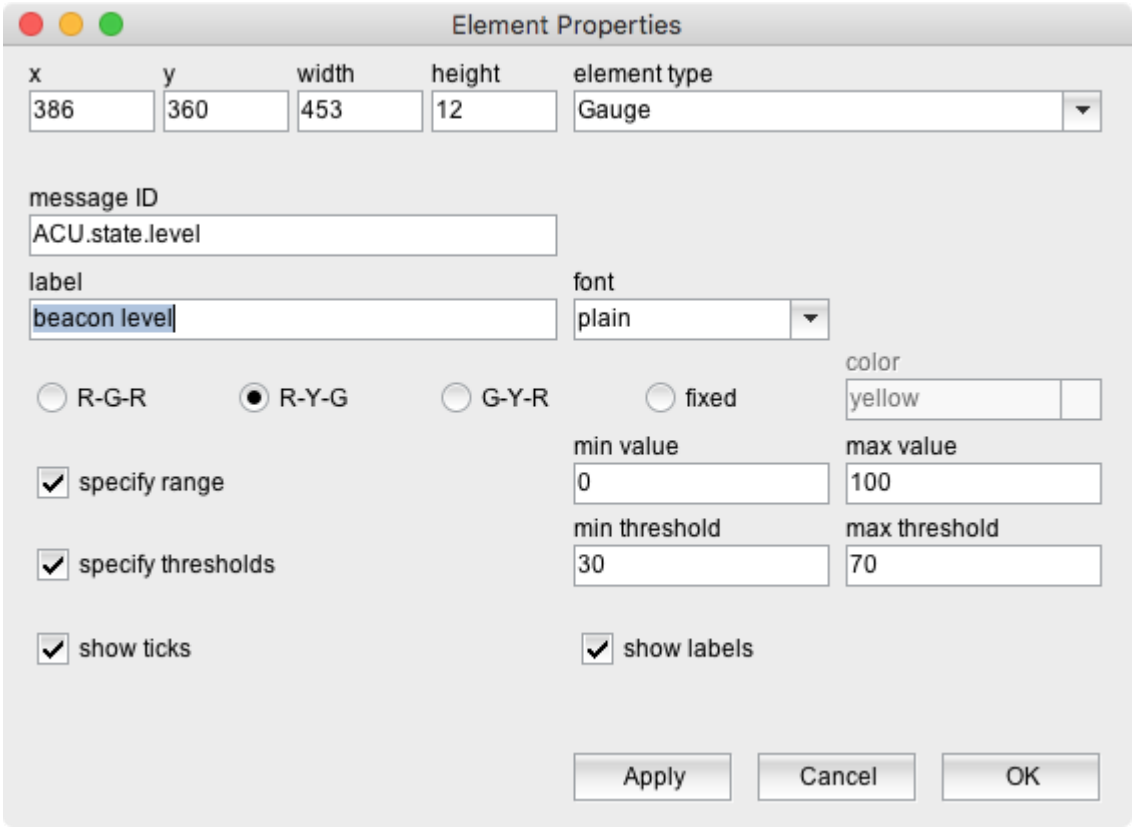
- message ID --- The message ID addresses the parameter the button shall control. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the ' Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- privilege level --- The privilege level necessary to access this button. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- Parameter buttons may be locked to disabled state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.

- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the button.
- use icon instead or label --- Check this mark in order to make the button show an icon instead of a label text.
- icon (up) --- The name of the icon to be shown in "up" state of the button. The field provides a list of all available icons / images on the sat-nms server.
- icon (down) --- The name of the icon to be shown in "down" state of the button. The field provides a list of all available icons / images on the sat-nms server.
- label (up) --- This label text is drawn on the button in "up" state.
- label (down) --- This label text is drawn on the button in "down" state.
- color --- The color attribute of a parameter element sets the color of the label shown on the button. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font to be used for the button label.
- query before applying changes --- Check this mark to make the element show a query before a parameter gets actually commanded.
- query text --- The query text to be shown in the pop-up window. you may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set.
- value (up) --- The value to be sent when the button is clicked an changes to "up" state.
- value (down) --- The value to be sent when the button is clicked an changes to "down" state.
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
    - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
    - To change the color assigned to a given parameter value, first select the table entry

of interest, change the color value and finally click "Set".
- To delete a value / color pair from the list, select the table entry and then click "Delete".

3.16.2.4.16 Gauge (Bar Graph)

The *Gauge* display element shows a numeric parameter value as a horizontal bar in an entry field like frame. The gauge element is capable to adjust the scale factor for the gauge automatically from the parameter's range definition. Alternatively the scale parameters may set explicitly.



The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses the parameter the element shall display. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the 'Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- label --- The label text drawn above the gauge / bar graph.
- font --- The font attribute applies to the label text shown above the gauge element.
- color scheme --- The gauge element knows three color schemes, allowing to change the bar color dynamically with the monitored value: | *color scheme* | *description* | | ---- | ---- | | R-G-R | RED/GREEN/RED The bar is displayed green while the value is between the minimum and maximum thresholds. If the value is outside the limits, the bar is displayed red. | | R-Y-G | RED/YELLOW/GREEN The bar is displayed red while the value is below the

minimum threshold value. It changes to yellow if the value is between the minimum and maximum thresholds. If the value exceeds the maximum threshold, the bar color changes to green. | | G-Y-R | GREEN/YELLOW/RED The bar is displayed green while the value is below the minimum threshold value. It changes to yellow if the value is between the minimum and maximum thresholds. If the value exceeds the maximum threshold, the bar color changes to red. | | fixed color | The bar is displayed with the color defined with the field 'color'. |

Please note, unless you specify min/max thresholds for the monitored value, only the fixed color mode is available.

- color --- The bar color used with the fixed color scheme. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- specify range --- Check this mark to define the range which is displayed by the gauge element.
- min value --- The minimum value of the displayed range.
- max value --- The maximum value of the displayed range.
- specify thresholds --- Check this mark to define the threshold values which are used for RGR and GYR color schemes.
- min threshold --- The minimum threshold value.
- max value --- The maximum threshold value.
- show ticks --- check this mark to make the element show scale ticks vor the min/max and threshold values below the gauge.
- show labels --- With this mark checked, below the gauge element labeled ticks are shown for the min/max and threshold values. (requires "show ticks"). Bar graph elements with this option enabled should be configured with a width that is sufficient to show the scale labels without overlap.

Scaling and Threshold Display

Principally there are three ways to use a gauge element and to control the scaling of the display:

1. The default behavior of the gauge element is to examine the display scaling from the range definition of the displayed variable. The scaling is set that the complete range of the variable is displayed. This works fine if the displayed variable comes with a properly set range definition. Some drivers however define the range of read only parameters as 0..0 (means 'don't know the limits') or as a extraneous wide range to avoid overflows. Variables with such range definitions will not display in a meaningful way. The default scaling does not provide any threshold monitoring or tick display.
2. You may explicitly specify the display range with a range definition in the appropriate fields as described above. Such a definition overwrites the default scaling of the gauge element. You may include unit descriptions to the scale values, but you should consider, that long scale labels tend to overlap and become unreadable.
3. If you let the gauge element monitor the 'gauge' status variable of aLevel-Set logical device, this state variable will provide it's own display range and threshold definition. This

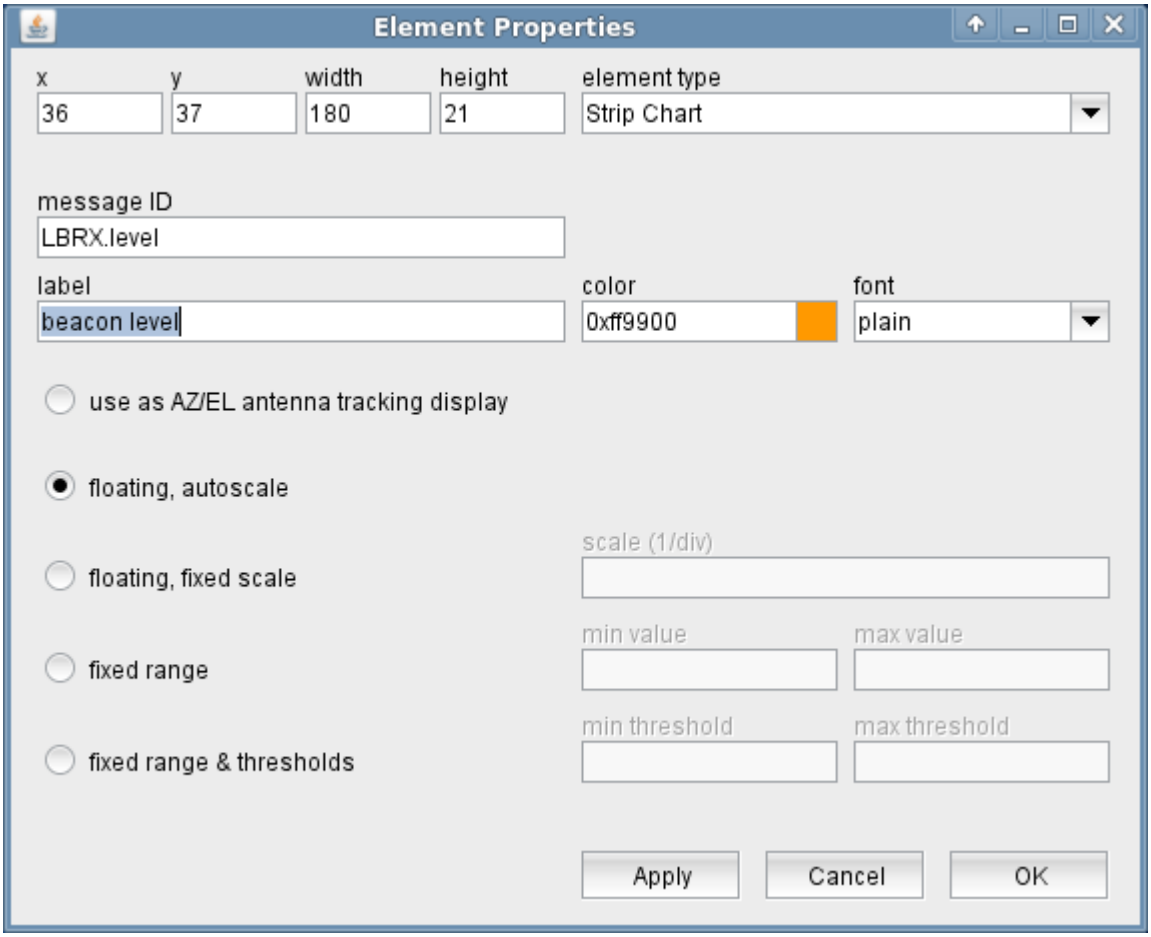
replaces the default scaling and also a scaling explicitly defined for this element. The display range provided by the Level-Set device always includes threshold definitions, but does not turn on the display of labeled ticks under the bar graph. If you want to enable this, you have to specify a dummy display range and thresholds in the fields described above.

3.16.2.4.17 Strip Chart

The *Strip Chart* display element shows a strip chart of a numeric parameter. The chart advances with a constant speed of 1 pixel / second. The default y-scale is 1/division but may be changed by clicking to the chart with the right mouse button.

By default, the strip chart element lets the y-scale offset follow the displayed value that the recent measurement samples are shown in the diagram. This behavior is optimized for applications where the strip chart shall indicate a 'trend' for the displayed value, using an element height of only 50 pixels or less.

Optionally you may disable this variable scaling by defining fixed scale values in the 'Text / Image Name' field. With this fixed y-axis scaling an additional threshold check may be added, turning the chart background to red if the actual value exceeds the defined limits. When used with limit checking, the height of the chart should be at least 100 pixels.



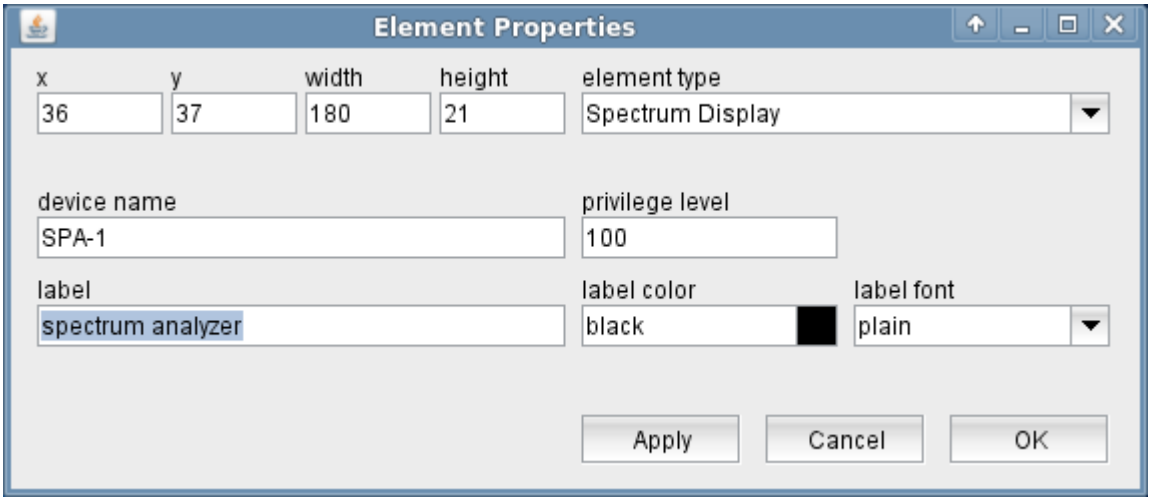
The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses the parameter the element shall display. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the 'Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- label --- The label text drawn above the strip chart
- color --- Defines the pen color and the color of the label drawn above the chart. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font attribute applies to the label text shown above the strip chart.
- display mode --- Selects the display mode of the strip chart element. Five modes are available:**use as AZ/EL antenna tracking display** The strip chart display element also may be used to place an Az/El diagram into a user interface for an Antenna-Tracking logical device. To do this, select this option and let the display element listen to the '.state.mode4' parameter of the Antenna-Tracking device. The Az/El diagram has a fixed size, you should align the display element size that the diagram is displayed properly.**floating, autoscale** This is the standard display mode. The strip chart aligns the reference value (the medium display line) in a way that every new display point is shown on this line. The initial scale is 1 unit/div.**floating, fixed scale** Works like the above mode, but the scale value gets initialized with a user defined value.**fixed range** The min/max values to display are defined explicitly, there is no automatic reference level alignment with this mode.**fixed range & thresholds** Like the above mode, but additionally there are min/max thresholds defined. If the actual value of the monitored parameter, the display background changes to red.
- scale --- The initial scale value used with the 'floating, fixed scale' mode.
- min value --- The minimum value to display for the fixed range display modes.
- max value --- The maximum value to display for the fixed range display modes.
- min threshold --- The lower threshold value for the "fixed range & thresholds" mode.
- max threshold --- The upper threshold value for the "fixed range & thresholds" mode.

3.16.2.4.18 Spectrum Display

The *Spectrum Display* element embeds the spectrum display of a spectrum analyzer device in the screen. The display may be scaled to an almost arbitrary size, however the following limitations apply:

- The annotation labels which reflect the actual spectrum analyzer settings are not scaled with the display element. If the element is very small, the labels may overlap and become unreadable.
- Resizing the spectrum display element is only possible in certain steps, the grid behind the spectrum curve is always made up of *10*n + 11* pixels in one dimension. The screen editor software knows about this, snaps the size of the element to the next (smaller) valid size after you release the mouse button.



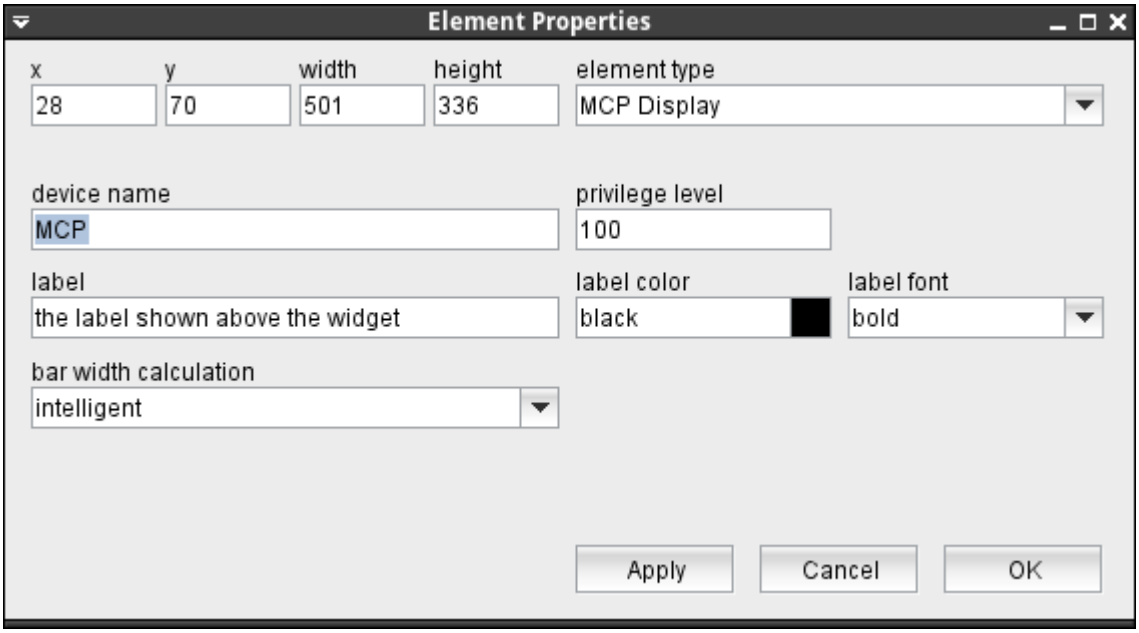
The attributes listed below may be configured to customize a display element of this type:

- device name --- The name of the spectrum analyzer device to operate.
- privilege level --- The privilege level necessary to operate the spectrum analyzer device.
- label --- The annotation label displayed above the spectrum display.
- label color --- The color of the label displayed above the spectrum display. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- label font --- The font attribute sets the font of the label shown above the element.

3.16.2.4.19 MCP Display

The *MCP Display* element embeds the measurement display of a spectrum analyzer device in multi channel power measurement mode into the screen. The display may be scaled to an almost arbitrary size, however the following limitations apply:

- The annotation labels which reflect the actual device settings are not scaled with the display element. If the element is very small, the labels may overlap and become unreadable.
- Channels may displayed very narrow if the display is too small, labels may become unreadable in this case.

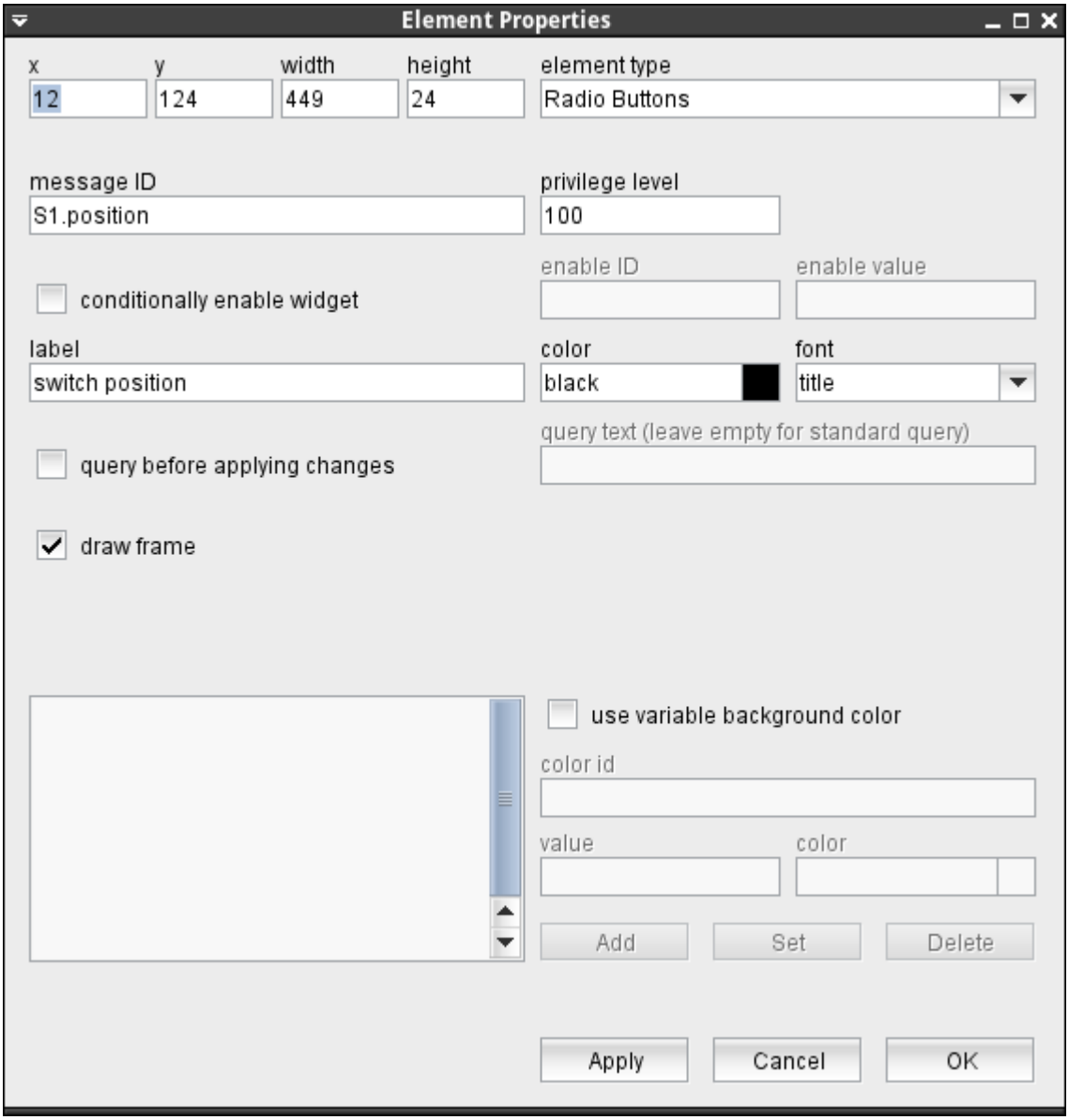


The attributes listed below may be configured to customize a display element of this type:

- device name --- The name of the spectrum analyzer device to operate.
- privilege level --- The privilege level necessary to operate the spectrum analyzer device.
- label --- The annotation label displayed above the spectrum display.
- label color --- The color of the label displayed above the spectrum display. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- label font --- The font attribute sets the font of the label shown above the element. This font also applies to the name labels shown below the channel bars.
- bar width calculation --- This parameter lets you select how the screen element calculates the width of the bars from the channels' bandwidth. 'all same width' draws all bars with the same width, regardless of the channels' bandwidth. 'strict proportional' calculates the width of the bars following the ratio of bandwidths - even if this leads to very narrow bars with unreadable figures. 'intelligent' which is the default, draws wide channels wider than narrow ones, but limits the ratio of bar widths to 1:3.

3.16.2.4.20 Radio Buttons

The *Radio Buttons* display element is a component to display and edit CHOICE type M&C parameters as a number of radio buttons. Depending on the height of the element, radio buttons are positioned in a row or in a column. In the screen editor, the parameter element is shown as a simple rectangle rectangle.



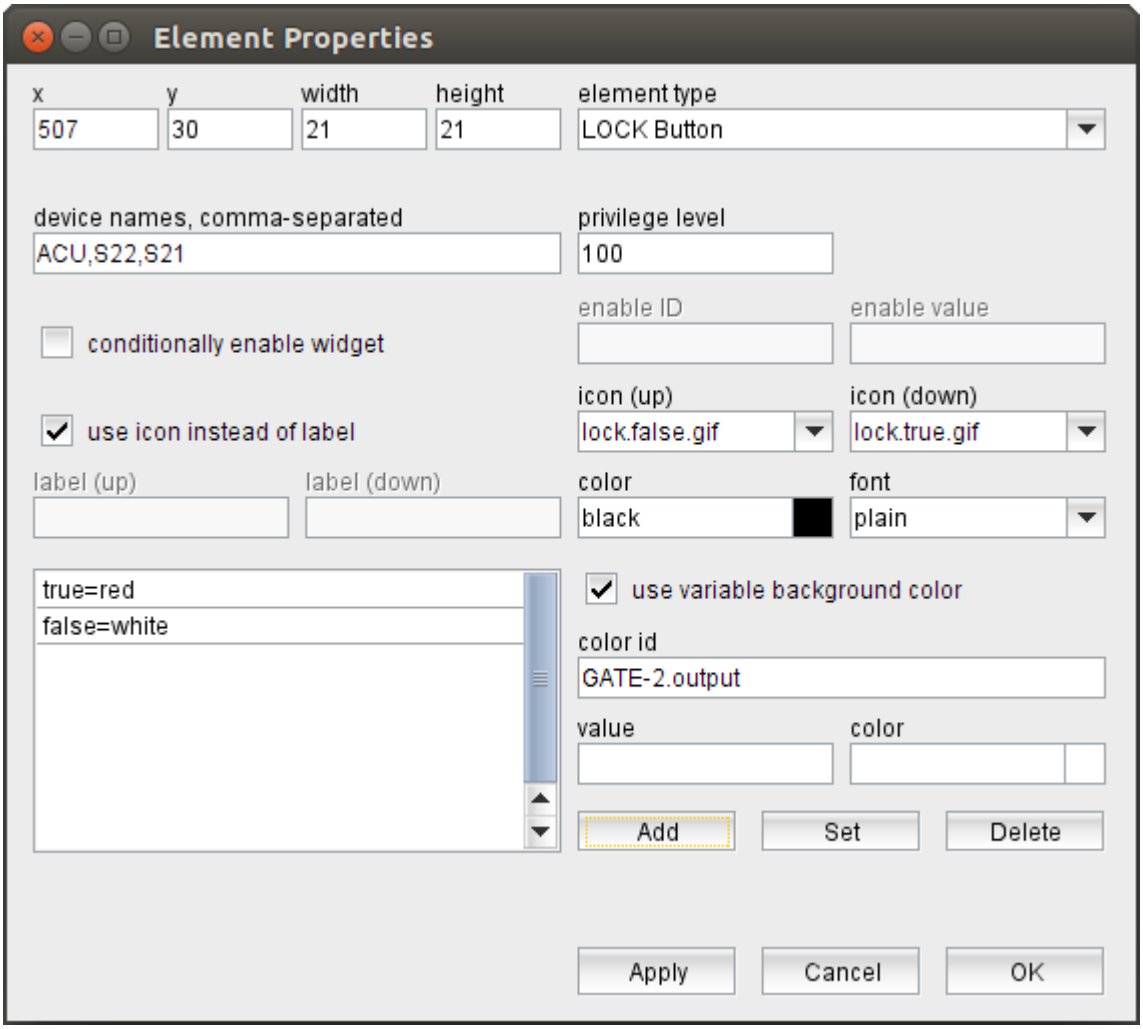
The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID addresses the parameter the element shall display or edit. The message ID consists of the device name followed by a colon and the name of the parameter. *XMIT-1.tx.on* for example addresses the variable *tx.on* at a device named *XMIT-1* . With the ' Device Driver Index ' in the appendix of this document there is a list of variable names provided for each device driver the software supports.
- privilege level --- The privilege level necessary to change this parameter. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- Parameter fields may be locked to read only state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.

- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the entry field.
- label --- The label text is drawn above the element.
- color --- The color attribute of a parameter element sets the color of the label shown above the entry field. The text in the parameter field itself always is shown in the default text color of the selected look & feel. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font attribute applies to the text shown in the parameter field itself. The label text is shown with the standard ("plain") font size for all font selections except the smallest one. In the latter case the label is drawn using this small font size, too.
- query before applying changes --- Check this mark to make the element show a query before a parameter gets actually commanded.
- query text --- The query text to be shown in the pop-up window. you may leave this field empty, the program uses a standard query text in this case.In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set.
- draw frame --- Checking this mark makes the element show a rectangular frame around the radio buttons.
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
  - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
  - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
  - To delete a value / color pair from the list, select the table entry and then click "Delete".

3.16.2.4.21 LOCK Button

The LOCK Button lets you lock the operation of a number of device devices that no other user can change device settings. The button is in pressed state if at least one of the devices it is assigned to is locked by some user. You only may realease the button if you set the lock by yourself of if you have a privilege level of 150 or more.



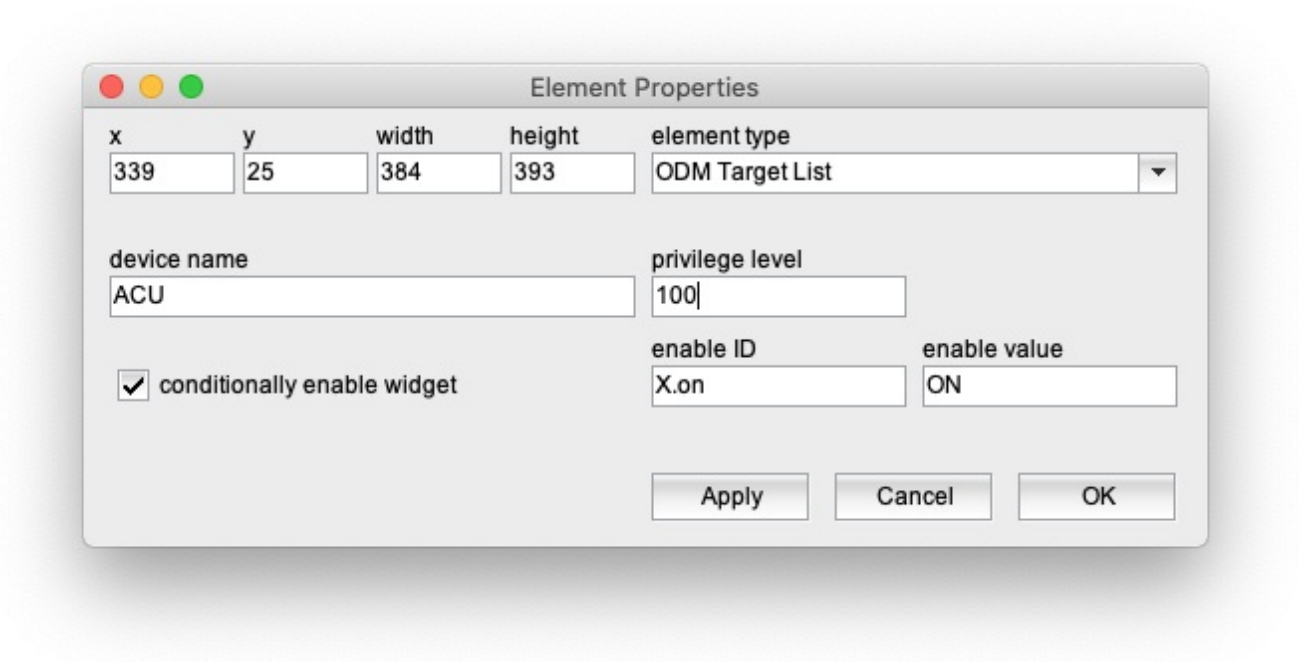
The attributes listed below may be configured to customize a display element of this type:

- device names, comma-separated --- As the field heading suggests, this field expects a comma-separated list of the devices, this button shall control.
- privilege level --- The privilege level necessary to access this button. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- Parameter buttons may be locked to disabled state unless another parameter matches a given value. For instance, changing the position of a wave guide switch may be inhibited while the carrier is switched on. Check this mark to activate this feature.
- enable ID --- This field defines the ID for the enabling parameter.
- value --- The value belonging to the parameter above. The actual value of the parameter

addressed by "enable ID" mut match this value in order to enable the button.
- use icon instead or label --- Check this mark in order to make the button show an icon instead of a label text.
- icon (up) --- The name of the icon to be shown in "up" (unlocked) state of the button. The field provides a list of all available icons / images on the sat-nms server.
- icon (down) --- The name of the icon to be shown in "down" (locked) state of the button. The field provides a list of all available icons / images on the sat-nms server.
- label (up) --- This label text is drawn on the button in "up" (unlocked) state.
- label (down) --- This label text is drawn on the button in "down" (locked) state.
- color --- The color attribute of a parameter element sets the color of the label shown on the button. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation. '#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.
- font --- The font to be used for the button label.
- use variable background color --- Checking this mark enables the variable background color feature for this screen element. Depending on the actual value of the variable addressed by 'color id', the element's background is set using the translation table shown in the lower left corner of the dialog. If the actual value of the color id variable does not match any of the table entries or if no color id is specified, the first color in the list is used to draw the element's background.The buttons Add/Set/Delete along with the value and color fields are used to edit the color translation table.
- color id --- The message ID of the parameter which controls the background color of this element.
- value --- The parameter value of the actually selected table entry
- color --- The color specification of the actually selected table entry. This may either be the name of a basic color (red, yellow etc.) or a hexadecimal RGB color specification like '#FF8000'. The latter defines some sort of orange, consisting of 0xFF red, 0x80 green and 0x00 blue. The actually selected color is shown in the right part of the entry field, clicking to this opens a color chooser dialog which lets you select / compose the the color to your needs.
- Add / Set / Delete --- These buttons control editing the color translation list.
    - To add a new value / color pair to the list, fill the value and color fields, then click to "Add".
    - To change the color assigned to a given parameter value, first select the table entry of interest, change the color value and finally click "Set".
    - To delete a value / color pair from the list, select the table entry and then click "Delete".

3.16.2.4.22 ODM Target List

The ODM Target List screen element lets you embed the list of targets of aSatService-ACU-ODM antenna controller in the screen. Targets may be recalled (which moves the antenna to the stored position and sets the tracking parameters associated with this target), saved or deleted.



The attributes listed below may be configured to customize a display element of this type:

- device name --- This parameter is interpreted differently depending on the context where the ODM Target List screen element resides:
  - When placed in a user defined screen or in the main screen of the application, 'device name' must be set to the name of the ODM device it shall refer to.
  - When placed in the device screen of another device which defines a configuration variable with the antenna controller device name, the name of this configuration variable must be entered as the 'device name'.
  - Finally, when used in the device screen of the ODM device itself, 'device name' must de set to '@'

Please note, that the ODM Target List screen element only works withSatService-ACU-ODM antenna controller device type, other antenna controllers are not supported.

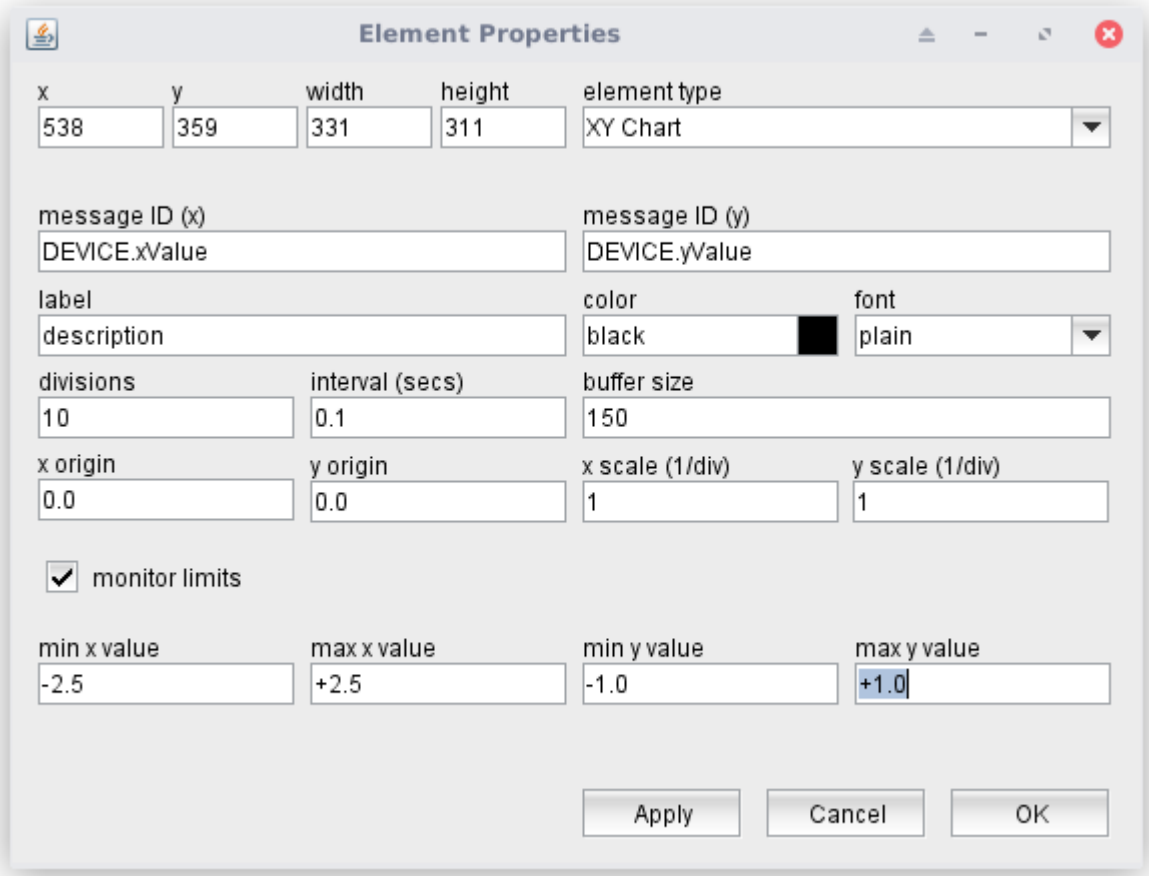
- privilege level --- The privilege level necessary to access this element. The predefined screens of the software use 100. In your own screens you may set certain parameters to higher privilege levels, limiting the group of operators permitted to change the parameter.
- conditionally enable widget --- The target list widget may be locked to disabled state unless another parameter matches a given value. For instance, a TX-ON state may disable the selection of another satellite by means of the target list widget.
- enable ID --- This field defines the ID for the enabling parameter. When used in a device screen, the ID is interpreted as a variable name of the device, this screen belongs to.
- value --- The value belonging to the parameter above. The actual value of the parameter addressed by "enable ID" mut match this value in order to enable the button.

3.16.2.4.23 XY Chart

This element shows the relation of two numeric variables in an X/Y diagram, featuring a 'track' which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

The X/Y chart display may be scaled to an almost arbitrary size, however the following limitations apply:

- The annotation labels which reflect the actual display settings are not scaled with the display element. If the element is very small, the labels may overlap and become unreadable.
- Resizing the X/Y chart display element is only possible in certain steps, the grid behind the curve is always made up of a number pixels in one dimension which is divisible by the configured number of diagram divisions without remainder. The screen editor software knows about this, snaps the size of the element to the next (smaller) valid size after you release the mouse button.



The attributes listed below may be configured to customize a display element of this type:

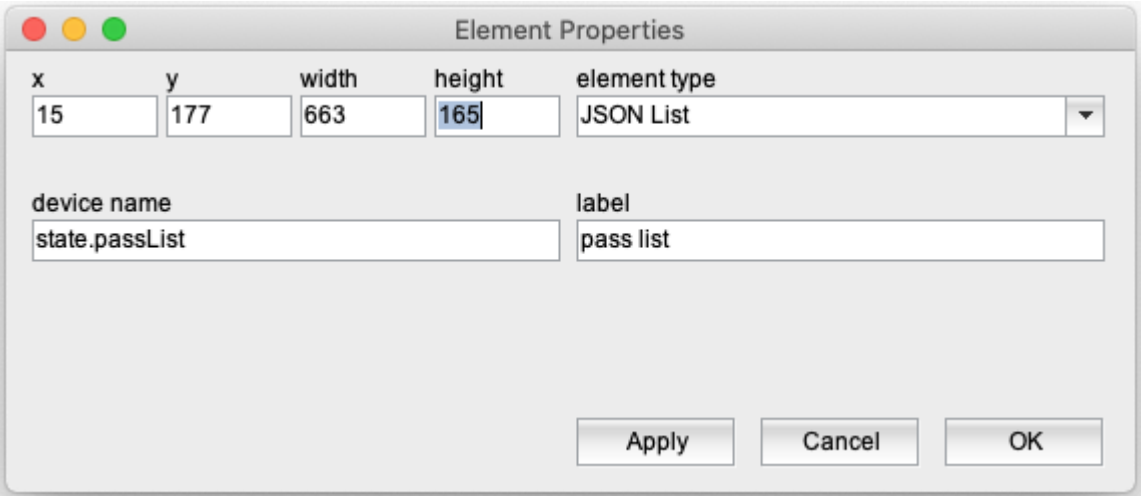
- messageID(x) message ID (y) --- The message IDs of the parameters to show.
- label --- The annotation label displayed above the X/Y chart
- color --- The color of the label displayed above the X/Y chart. Colors may be specified by standard color names like red / green / blue or using a hexadecimal RGB notation.

'#FFFF00' for example specifies yellow. You may click to the color example field in order to open a color chooser dialog and select the color from there.

- font --- The font attribute sets the font of the label shown above the element.
- divisions --- The number of divisions shown in the diagram for both directions. You may enter arbitrary values here, but to achieve some readability, you are encouraged to us common values like 2 ('hair cross'), 4 or 10.
- interval --- The update time interval for the display in seconds. 0.1 means to add every 100msec a new value to the display and remove the oldest value from the buffer at the same time.
- buffer size --- The display maintains a "first in first out" buffer of a size defined with this parameter. The buffer provides a short time memory the display shows as a trace of past values. Values in the range 100-300 are a good choice, larger values may slow down the client application. Please note, that the buffer gets cleared when the window is closed.
- x origin y origin --- The origin values refer to the*center* of the diagram. Using an even number of divisions helps to identify the diagram center as the crosspoint of the middle grid lines
- x scale y scale --- The scale values are per division. with xorigin = 0.0 and x scale = 1.0 an x value of 1.0 gets displayed one division right of the diagram center.
- monitor limits --- With this mark checked, the display monitors the*actual* X/Y values to be within the limits defined below. The limit values are shown as a dark red rectangle. When the while the actual X/Y values exceed the limits, the disgram background becomes red.
- min x value max x value min y value max y value --- The limit values for X and Y to be checked.

3.16.2.4.24 JSON List

The JSON List element is a special screen element required for some devices which distribute a state table/list in JSON format. It can only be used for this purpose.

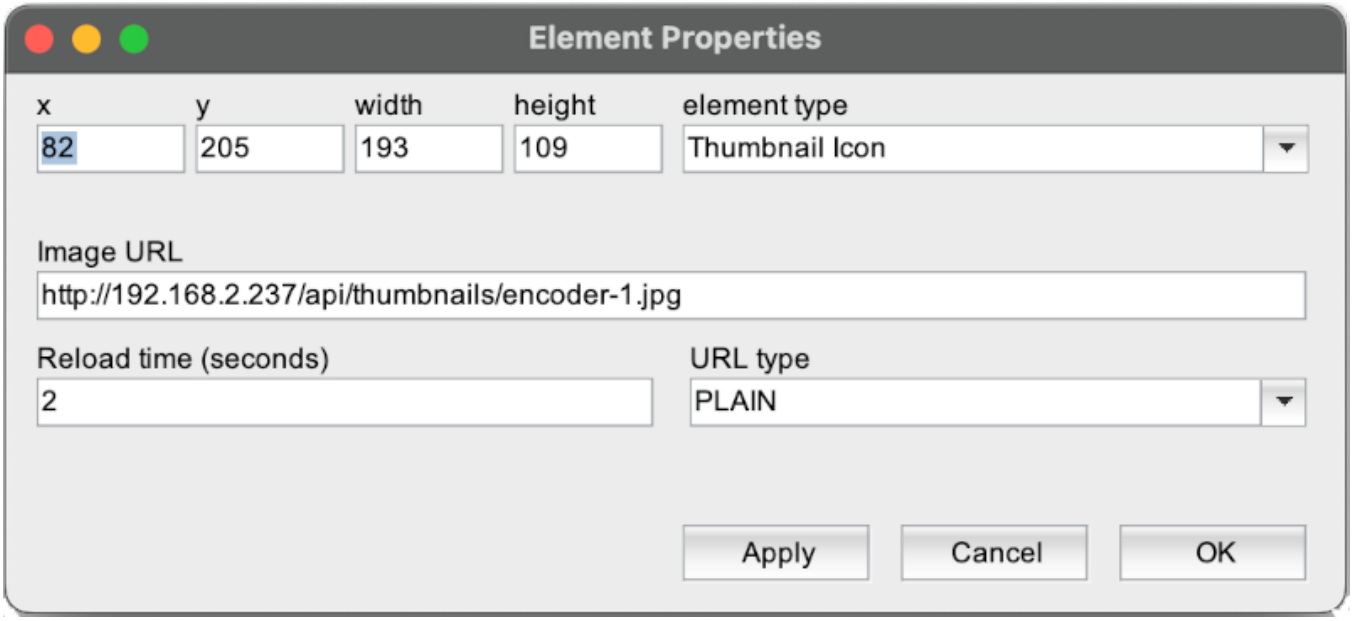


The attributes listed below may be configured to customize a display element of this type:

- message ID --- The message ID of the JSON list to show.
- label --- The annotation label displayed above the element

3.16.2.4.25 Thumbnail Icon

The Thumbnail Icon element lets you display a thumbnail of the video actually processed by a device lika an encoder, decoder or gateway. The displayed icon is a static image, updated every couple of seconds.



The attributes listed below may be configured to customize a display element of this type:

- Image URL --- The URL where to fetch the image.
- Reload Time --- The interval when the image shall be reloaded (seconds)
- URL Type --- One of PLAIN or ADVANCED. With PLAIN the URL is used as it is, with ADVANCED CGI parameters for the actual time and the image width and height are appended to the URL.

REMARKS

- The device to get the thumbnail images from must support this feature.
- The device to get the thumbnail images from must be accessible in the network from the client's point of view.
- Thumbnail Icons are - when resized by mouse drag in the GUI editor - forced to 16:9 ratio. If another image ratio is required, image width and height must be entered in the dialog shown above.
- in PLAIN mode, the image received from the device gets scaled to fit the screen element size, in ADVANCED mode no scaling is done as the software queries the image already in the correct size from the device.

3.16.2.5 Configuring a Tabbed View Client

Configuring a Tabbed View user interface is mainly a job of planning and organization. The layout of navigation bar buttons and VLC screens ist defined in a configuration file, but before editing this file some preparation have to be done:

1. Define, which views / tabs shall be present in the Tabbed View Client
2. For each tab define which VLC screens shall be contained. Edit the screens using the sat-nms Screen Editor Window .
3. Edit the Tabbed View configuration file to put things together.

The NMS directory tree provides an own subdirectory named "tviews", containing the configuration files for the Tabbed View Client. The file called "default" is used by the client, unless the client is forced to read another file (that's a command line option of the client program)

Configuration file format

The configuration file is a free form text file defining the tab layout with a couple of keywords. In the file, whitespace and C/C++ style comments are ignored. Syntax and keywords are best explained using this example:

```
// example tabbed view definition file
//

SHOW-FAULTICON NO
SUPPRESS-OFFLINE-FAULTS YES
SPACING 3

BUTTON "TAB 1"

   SCREEN vlc0001 vlc0001_rx_Tabscreen
   SCREEN vlc0002 vlc0002_rx_Tabscreen
   SCREEN vlc0003 vlc0003_rx_Tabscreen
   GROUP G1

GAP

BUTTON "TAB 2"

   SCREEN vlc0003 vlc0003_tx
   SCREEN vlc0002 vlc0002_tx
   SCREEN vlc0001 vlc0001_tx

DEFAULTS

   USER u1 "TAB 2"
```

The file starts with two global definitions:

- SHOW-FAULTICON --- Must be followed by YES or NO. Defines if the main toolbar shall include a system wide "new fault" indicator. Depending on the type of usage of the equipment managed by the NMS, the system wide fault indicator may be red all the time, irritating the operators. In this case you may want to suppress the indicator by specifying

"SHOW-FAULTICON NO". If you omit the SHOW-FAULTICON definition, the Tabbed View Client shows the fault indicator.
- SUPPRESS-OFFLINE-FAULTS --- Must be followed by YES or NO. Defines if faults of an off-line VLC are signalled with a red lamp in the Tabbed View's navigation bar:
  - With NO, an off-line VLC is shown with a red lamp if the VLC signalled a fault when it was on-line the last time.
  - With YES, of-line VLCs always are shown with s gray lamp.

Default is NO, without the SUPPRESS-OFFLINE-FAULTS definition the software shows off-line VLCs with a red lamp if there is a faul known for the VLC.

- SPACING --- Must be followed by a number in the range 1..100. The SPACING value defines the room in pixels, the program reserves above and below each button. Hence, the space between two buttons is two times this value. The default for SPACING is 10 pixels.

The sequence of these definitions doesn't matter, however they should appear before the first BUTTON definition.

The BUTTON keyword defines a button in the navigation bar. It must be followed by the button label, enclosed in double quotes. Buttons are identified by their label, hence button labels should be unique within a Tabbed View definition.

The BUTTON definition also starts the definition of the tab / view assigned to this button. Within the tab definition, the following definitions are expected:

- GROUP *group-name* --- The GROUP definition is optional and may appear only once. Specifying a group for a button restricts the access to this tab to users being part of the specified user group. Omitting the GROUP definition for a button lets all (loggen in) users access this tab.
- SCREEN *vlc-id screen-name* --- The SCREEN clause defines one VLC screen appearing in the tab. The keyword SCREEN must be followed by the ID of the VLC to connect to (using the notation vlc####) and the name of the screen definition to display. The screen definition file is expected in the 'screens' directory, the file name must not contain any whitespace.There may be multiple SCREEN definitions for one tab, the screens are layed out one below each other in the order they appear in the configuration file. The number of screens actually is not limited, but it makes no sense to define more screens than fitting into the fullscreen application window. There are no scrollbars provided to manage a virtual screen larger than the physical screen size.

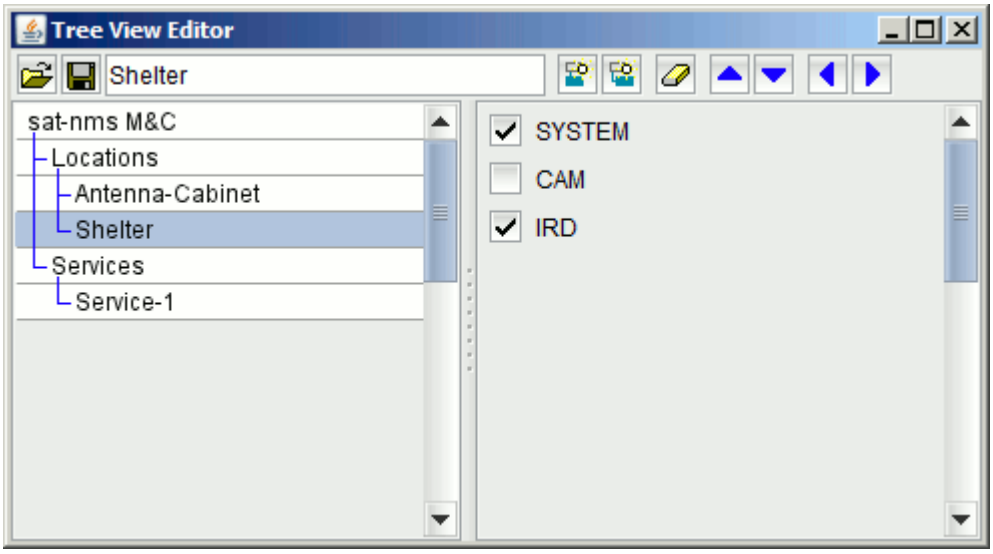
In the sequence of BUTTON definitions the keywords GAP and LINE may be used instead of a button clause. As GAP suggests, this tells the software to place a gap (10 pixels height) between two buttons in the navigation bar. LINE draws a horizontal line to separate buttons.

The BUTTON definitions optionally may be followed by a list of default screen assignments, preceded by the keyword DEFAULTS. Each default screen definition consists of the keyword USER, folled by a user name and the label of a button defined above. The button label must be enclosed in double quotes.

Defining a default tab for a certain user means to tell the software that this tab shall be opened automatically when the user logs on. For each user only one default tab may be defined. Together with the autologin feature available as command line option when the Tabbed View gets invoked, the program may configured to show one tab automatically when started.

3.16.2.6 Configuring the M&C Tree View

The Tree View Editor window lets you define or modify the tree / subsystem hierarchy used by the M&C Tree View. You open the Tree View Editor by clicking the appropriate tool bar button of the Tree View.



The left part of window shows the subsystem tree structure, the right part a list of all devices configured in the M&C system. Editing / modifying the tree structure is straight forward:

Selecting a tree node with the mouse shows the name of this node in the text field in the tool bar and at the right side places a checkmark in front of all devices which are attached to this node. Checking / unchecking devices attaches or detaches the to/from the actual tree node. Using the toolbar buttons you may add or remove tree nodes and move them up or down:

- --- Re-load the tree definition from its disc file, thus reverting all changes to the last saved state.
- --- Stores the changes made at the tree at the server and updates all Tree View windows actually open.
- --- Shows the name of the selected node. The name may be changed by editing the field and pressing RETURN. Please note, that changing a node's name will re-reference the node from its user screen and also will change the message IDs of the subsystem fault message of this node ans all its children.
- --- Adds a new node to the tree. The new node will be added as the last sibling of the selected node.
- --- Adds a new node to the tree. The new node will be added as the last child of the selected node.

- --- Removes the selected not an all its children from the tree.
- --- Moves the selected node one position up in the list of its siblings. Does nothing if the note is already at the topmost position.
- --- Moves the selected node one position down in the list of its siblings. Does nothing if the note is already at the lowest position.
- --- Moves the selected note on level up in the tree hierarchy. The highest level a node may be of is one below the root node.
- --- Moves the selected node one level down in the tree hierarchy. The node must have at least one sibling for this.

The nodes of the tree view in the sat-nms software also can be seen as subsystems of the M&C. The M&C server creates a fault flag for wach tree view node / subsystem which is defined using the Tree View Editor.

Subsystem fault flags summarize the fault states of all devices and subsequent tree nodes which are children the particular node. Like the summary fault of a device, the subsystem fault is one of "OK.", "WARNING" of "Summary FAULT". It states the most severe fault state one of the summarized devices or subsystems reports.

The subsystem fault flags can be accessed for display at the GUI through their parameter ID, which is defined as "SUBSYSTEM. ... .fault", where the three dots are to be replaced by the fully qualified name of the tree node. If for example a tree node "/Locations/Shelter-B" exists, the fault ID for this node is "SUBSYSTEM.Locations.Shelter-B.fault".

4 Administration

This section of the manual gives some information how to administer and configure the various components of the satnms M&C software. It describes how to invoke the programs used by the software and which command line parameters these programs understand.

Chapter Configuration Files contains a reference to all configuration parameters which can be defined in varipus configuration files of the software.

Chapter Event Database Configuration and Event Notification Add-Ons describes how to configure the event database and opetional add-ons which may be configured to trigger some actions on certain events.

Finally, the chapter Remote Control describes the features built into the satnms M&C software for being remotely monitored and controlled by an external software.

4.1 Program Invocation

The satnms M&C software consists of a number of programs running independently on the same or on separate machines. The following chapters describes how to invoke these programs and which command line parameters they accept.

4.1.1 M&C Server

The M&C server is run as a system service and is automatically started when the operating system comes up. Normally there is no need to change the command line parameters which are passed to the program by the the startup scripts. Most parameters of the M&C server may be configured by editing the vlc.propertis configuration file.

If the M&C server must be stopped or started explicitly, the follwing commands sould be used:

```
$ systemctl stop satnms
```

and

```
$ systemctl start satnms
```

respectively. To start or stop the M&C one must be logged in as 'root' on the machine hosting the M&C server.

4.1.2 Backend Server

The backend server is run as a system service and is automatically started when the operating system comes up. There are no command line parameters for the server program itself, the behavior of the backend is entirely controlled by the backend.properties file.

If the backend server must be stopped or started explicitly, the follwing commands sould be

used:

```
$ systemctl stop backend
```

and

```
$ systemctl start backend
```

respectively. To start or stop the backend one must be logged in as 'root' on the machine hosting the backend server.

4.1.3 Client Programs

The following paragraphs list the command line invocation syntax and options for the sat-nms client software. Being pure Java applications, the sat-nms clients may be run on severals operating systems. The explanations given in the following are mostly independent from the operating system you are using. For all examples given below, we assume that the file "client.jar" is located in your current working directory.

M&C Client

The M&C client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.MCFrame [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are

shown centered at this point.

- -f 0 -f 1 --- Starts the client in full screen mode. The main window covers the complete screen, no window decorations are shown. *-f 1* shows the satnms toolbar (to launch the event log for example) *-f 0* suppresses the tool bar.The full screen mode is intended to be used with touch screen panels. There are limited capabilities to handle child windows in *-f 0* mode, depending on the operating system of the client pc.
- -t --- Starts the client without the toolbar.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.
- -s <screen-name> --- Tells the program to load the given screen definition file instead of the "default" screen on startup. Screen definition files are expected in the "screens" subdirectory of the server. The -s also suppresses the display of the applications's splash screen on startup.
- -v --- Suppresses the program version match check on startup
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.
- -y <layout-name> --- Tells the program to load the given window layout instead of the layour "default".

TreeView Client

The TreeView version of the M&C client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.TreeView [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are shown centered at this point.
- -f 0 -f 1 --- Starts the client in full screen mode. The main window covers the complete screen, no window decorations are shown. *-f 1* shows the sat-nms toolbar (to launch the event log for example) *-f 0* suppresses the tool bar. The full screen mode is intended to be used with touch screen panels. There are limited capabilities to handle child windows in *-f 0* mode, depending on the operating system of the client pc.
- -t --- Starts the client without the toolbar.
- -o --- Enables automatic pop ups of the fault list window with every new fault.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run

are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.

- -s <screen-name> --- Tells the program to load the given screen definition file instead of the "default" screen on startup. Screen definition files are expected in the "screens" subdirectory of the server. The -s also suppresses the display of the applications's splash screen on startup.
- -v --- Suppresses the program version match check on startup
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

Tabbed View Client

The Tabbed View NMS client program is invoked by a command line looking like this:

java -cp client.jar satnms3.gui.TabbedView [options ...] <IP address of the server>

The IP address of the NMS server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -m x0,y0,x1,x2,xc,yc --- Specifies the (overall) bounds of a multi monitor setup. When operated on a single monitor PC, the client automatically determines the screen size and ensures that automatically placed windows remain within the bounds of the screen. The -m option is not necessary in this case. With multiple monitors, the abolute bounds of the rectangle enclosing all monitors must be specified with the -m option. -m must be followed by a space character and 6 numbers separated by comma characters: x0 = x coordinate of the top left corner y0 = y coordinate of the top left corner x1 = x coordinate of the bottom right corner y1 = y coordinate of the bottom right corner xc = x coordinate of center position yc = y coordinate of center position Example: 0,0,3840,1080,2880,540 describes a layout with two full HD monitors located side by side. The center position is defined as the center of the right monitor (the splash screen and some other windows are shown centered at this point.
- -s --- Suppresses the beeper ("silent").
- -t <tab-definition> --- Tells the program to load the given tab definition file instead of the "default" definition on startup. Tab definition files are expected in the "tviews" subdirectory of the NMS server.

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

Service Client

The ServiceClient client program is invoked by a command line looking like this:

java -cp client.jar satnms3.sclient.ServiceClient [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.
- -s <service-definition> --- Tells the client to use another service definition file than 'default.xml'. The service definition file must be placed in the ~/sclient directory at the M&C server
- -o --- Enables automatic pop ups of the fault list window with every new fault.
- -r <seconds> --- Enables the automatic reconnect feature. Should not be used directly,

the MasterClient application makes use of this switch.

- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

VioIP Client

The VioIPClient client program is invoked by a command line looking like this:

java -cp client.jar satnms3.vioip.VioIPClient [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -g <group-ip> --- Defines the group to be displayed by this client instance. Expects the group ID (not the name) as a parameter. This parameter is mandatory to make the program work, the group ID cannot be set or changed at the GUI.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -n 1 -n 2 -n 3 --- Tells the client how to treat "new faults". Without the -n option, new faults are not reported separately. The fault indicator in the main window's toolbar ia active while the server's summary fault is active.
    - **n1** --- The program treats new faults separately. A fault is new until an operator acknowledges it by clicking into the fault indicator or by acknowledging the fault in the event log or event report window. The new fault is displayed as a red mark labeled "NEW FAULT". If the fault disappears without being acknowledged, a gray "NEW FAULT" mark is shown.
    - **n2** --- Behaves like **-n 1** , but makes the program ring the keyboard bell while a new fault is pending.
    - **n3** --- Works like the previous, but suppresses the very first "NEW FAULT" after the start of the client. New faults which occurred while the client program did not run are not signalled, nor are gray "NEW FAULT" conditions. Only new faults recognized while the client program runs are shown.
- -q --- Suppresses some alerts like 'connection broken' and others.

- -r <seconds> --- Enables the automatic reconnect feature. Should not be used directly, the MasterClient application makes use of this switch.
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

VioIP Configurator

The VioIPConfigurator program usually is invoked with a button click from the VioIPClient program. Beside this, it also may be invoked as a stand alone program by a command line looking like this:

java -cp client.jar satnms3.vioip.VioIPConfigurator [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation provides name resolution for this address.

The following command line options may be used with the program:

- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -s --- Suppresses the splash screen.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -q --- Suppresses some alerts like 'connection broken' and others.
- -d <r,g,b> --- Sets the foreground color of disabled texts / elements in the GUI. r,g,b are numbers in the range 0..255, comma separated, specifying the RGB components of the color.
- -v --- Suppresses the program version match check on startup
- -V --- Tells the program to be verbose on the command line. This option is for debugging only, **never** set it for operational use.

CDB Table Editor

The CDB Table Editor program usually is invoked with a button click from the M&C Client program. Beside this, it also may be invoked as a stand alone program by a command line looking like this:

java -cp client.jar satnms3.gui.CDBTableEditor [options ...] <IP address of the server>

The IP address of the M&C server may be replaced by a host name if your network installation

provides name resolution for this address.

The following command line options may be used with the program:

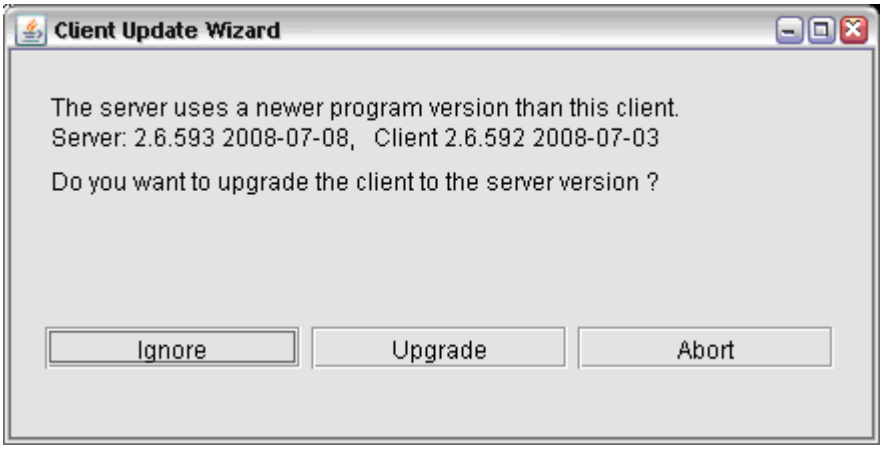
- -u <user-name> --- Specifies the user name for autologin. Must be used together with -p
- -p <password> --- Specifies the password for autologin. Must be used together with -u
- -s --- Suppresses the database selector widget in the toolbar.
- -l <look-and-feel> --- Lets you set the "look and feel" of the user interface to one of several definitions. Accepted values are: Aero, Fast, Aluminium, McWin, Luna, Pulsar, Noire, Mint, Graphite, Smart, Bernstein, Acryl and HiFi.
- -db <db-name> --- Selects the database to be edited. Default is 'channels'.

4.1.4 Automatic Software Update Of The Client

The sat-nms Client Software (version 2.6.591 and above) includes an automatic check for updates. After startup the client connects to the NMS server and compares the software versions. Client and server should always run the same version of the sat-nms software. The software update feature also supports a downgrade of the client software, e.g. if the server has been restored from a backup with an older software version.

Upgrade

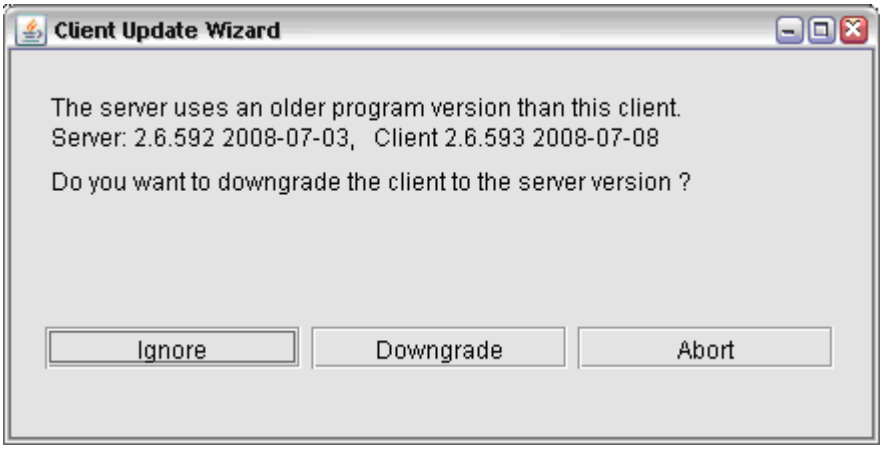
If the NMS server is running a newer version of sat-nms you can upgrade the local client to this version.



- Ignore: --- This ignores the version discrepancy and you will working with the already installed client program.
- Upgrade: --- The newer client program will be installed on the local machine
- Abort: --- Aborts the client program, cancels the program download

Downgrade

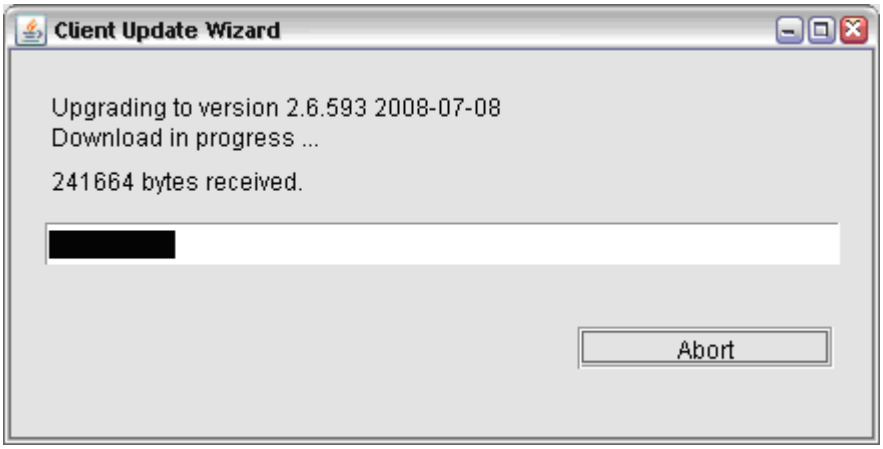
If the NMS server is running an older version of sat-nms you can downgrade the local client to this version.



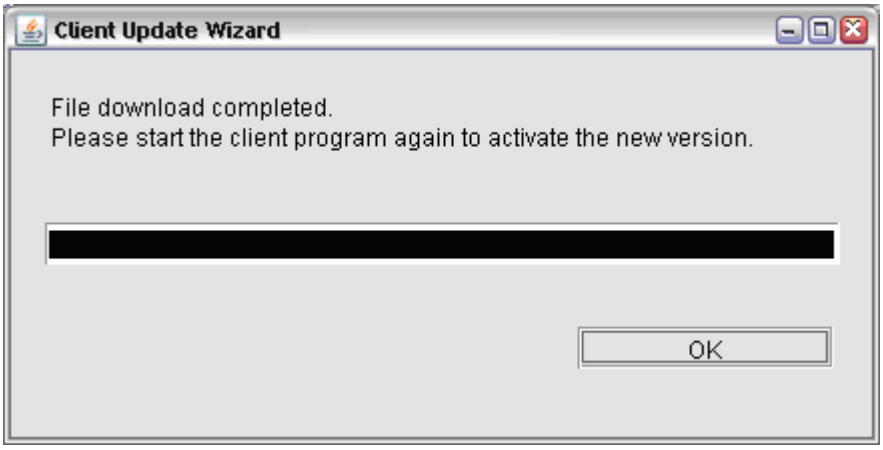
- Ignore: --- This ignores the version discrepancy and you will working with the already installed client program.
- Downgrade: --- The older client program will be installed on the local machine
- Abort: --- Aborts the client program, cancels the program download

Download process

After pressing the upgrade or downgrade button, the software starts the download of the client program from the server to the local machine. With a LAN connection this takes only a few seconds, if you are connected through a dial up connection may take somewhat longer.



You may cancel the download at any time by clicking the 'Abort' button. After successfully downloading the software displays a message and after selecting OK the client program will be closed.



Now simply restart the client again as described in chapter 'start client' and you will work with the upgraded or downgraded version.

4.2 Configuration Files

The satnms M&C software provides a number of configuration files which control wide aspects of the functionality of the software files all use an easy to read and mostly self explaining format. The files explained in the following chapters are:

- backend.properties: This file is located in the base directory of the satnms backend installation. It controls the behavior of the backend server.
- user-ip-restrictions.json: This file is also located in the base directory of the satnms backend installation. It controls which user may login to the backend from which machine.
- vlc.properties: This file is located in the base directory of the satnms M&C installation. It controls all important parameters of the M&C server.
- client.properties: This file also is located in the base directory of the software. It is read by every legacy Java client connecting to the server. It controls important aspects of the client software's behavior.
- satnms.rc: Controls the startup of the satnms software and programs which have to be initialized before the M&C server itself can be started.
- system.name: Defines the name of the M&C system
- traprec.json: Controls the behavior of the SNMP trap receiver built into the software.

4.2.1 backend.properties

The backend.properties file controls the configuration of the backend server. How many and which M&C systems are controlled by the backend, how to authenticate at the backend and some debugging options are defined in this file. The backend.properties file is read once by the software on startup, so any changes to this file become effective with the next start of the backend. The backend installation package contains a backend.properties.default file which can be used as a template to setup a new properties file from scatch

Controlled M&C systems

For every M&C system controlled by the backend, at least two parameters must be set: The M&C system's IP address (`backend.mnc.*.ip`) and it's name (`backend.mnc.*.name`). Example for the first M&C:

```
backend.mnc.1.ip=127.0.0.1
backend.mnc.1.name=MNC
```

M&Cs are numbered 1,2 ... n, number 1 is called the primary M&C. It must always exist, the backend uses it to get screens, driver definitions and to authenticate users during the login. The name of each M&C should match the name which is set at the M&C server itself. This ensures that the M&C name presented at the UI frontend matches the M&C name appearing in event log messages.

The backend supports redundant M&C pairs, monitors and control the switchover. To define a redundant M&C pair, some more parameters are required. They are the IP address of the backup M&C and some timing parameter controlling the switchover process.

```
backend.mnc.2.ip=192.168.2.229
backend.mnc.2.name=MNC-2
backend.mnc.2.backup.ip=192.168.2.227
backend.mnc.2.backup.retries=3
backend.mnc.2.backup.delay=15
```

As soon as `backend.mnc.*.backup.ip` is defined for a M&C, this is considered to be a redundant pair. The `backend.mnc.*.backup.retries` parameter defined how many reconnect attempts to the primary M&C shall be made until a switch over is done. `backend.mnc.*.backup.delay` specifies the wait time between connection attempts.

EventDB server

The backend needs to know where the event database is hosted. The IP address is specified with:

```
eventdb.server=127.0.0.1
```

For small installations this is typically the same machine as the backend

SQL database access

The backend stores information about satellites, satellite channels and antenna pointings in SQL a database. The setings below specify how to access this database. Again in a small installation the database will run on the same host as the backend itself.

```
db.satlist.host=127.0.0.1
db.satlist.user=satnms
db.satlist.password=satnms
db.satlist.usessl=false
```

The backend may use a redundant database configuration where the primary database server is configured r/w and the backup server is r/o. For such a configuration the `db.satlist.host` parameter defines the primary and backup server addresses separated by a comma. The first address denotes the primary database server.

```
db.satlist.host=192.168.0.100,192.168.0.101
db.satlist.keep=300
```

The `db.satlist.keep` parameter defines the time (secs) after which the backend cuts a connetion to the backup server in order to try if the primary one is available again.

Inventory database

The inventory database is part of the SQL database which also stores satellite channels etc. The inventory feature must be enabled explicitly by setting the parameter below to true:

```
db.inventory.enabled=false
db.inventory.debug=false
```

The `db.satlist.*` parameters shown in the section above this must also be setup properly to enable the inventory database. Setting the debug flag *true* lets the software print out all SQL traffic regarding the inventory into the .panic.log file. This causes huge log files, therefore shouls be used only investigating problems, not during normal operation.

OAuth2 authentication

The backend uses OAuth2 to authenticate API calls. it provides a local authority which issues tokens for local logins. The parameters below configure the behavior of this feature

```
oauth2.access.token.valid.time=3600
```

Specifies the time a token is valid (seconds). default is 3600

```
oauth2.refresh.token.valid.time=7200
```

Specifies the time the refresh token is valid (seconds). default is 7200

```
max.sessions.total=0
```

Specifieds the max. number of session which totally may be open at the same time (default 0, means don't care)

```
max.sessions.user=0
```

Specified the max. number of session which may be open at the same time for the same user (default 0, means don't care)

The backend accepts JWT tokens from an external authentication authority if the following parameters are set URL to access the JWKS from this authority. default is empty.

```
oauth2.jwk.provider=
```

The parameter below secifies IP address of an HTTP(s) proxy if the JWKS provider shall be accessed thru this proxy. default is empty (no proxy)

```
oauth2.jwk.proxy=
```

The parameter below specifies the AD group name to privilege level look up list. It consists of group-name:privilege-level pairs, separated by comma characters. default is an empty list.

```
oauth2.groups=SATNMS-ADM:150,SATNMS-OP:110
```

Beside from the "Authentication" header the backend may read the JWT token from a cookie coming with the API call. default is mpty (don't accept JWT from a cookie)

```
oauth2.cookie.name=
```

LDAP local login

For local logins the backend may look up the credentials of a user at an LDAP directory server instead of parsing the `.users` file from the primary M&C. The settings below are used to specify how to access the LDAP server and how to interprets the data read from there:

```
ldap.enable=yes
ldap.url=ldaps://ad.mydomain:636
ldap.default.domain=ad.mydomain
ldap.search.user=searchsatnms
ldap.search.password=searchsatnms
ldap.groups=SATNMS-ADM:150,SATNMS-OP:110
ldap.keystore.name=ldap.p12
ldap.keystore.password=satnms
```

Debug settings

The backend offers a number of parameters which enable extensive logging of certain events, internal messages of parameter updates for debugging purposes. For normal operation you should keep the disabled all because log files can become very large with debuggin enabled. Debugging also may reduce the performance of the backend.

The parameter/status communication to a specific M&C may be logged by setting this parameter true (replace '*' with the index of the M&C to switch to verbose.

```
backend.mnc.*.verbose=true
```

Setting this true logs every message sent over a websocket. Creates a huge amount of log data, logs are better to interpret if only one single frontend with one window is connected to the backend

```
websocket.verbose=false
```

Setting the parameter blow to true logs m&c connect/disconnect events.

```
websocket.logConnected=false
```

Setting the parameter below to true logs the oauth2 authentication process for every API call in detail.

```
oauth2.verbose=false
```

The parameter below controls logging for LDAP logins. Please note that this setting expects 'yes' to be activated.

```
ldap.verbose=no
```

4.2.1.1 Example configurations

MNC and Backend services on same host (all-in-one installation)

This is a very simple configuration running backend and MNC service on the same host without any additional features:

```
# MNC server
backend.mnc.1.ip=127.0.0.1
backend.mnc.1.name=MNC

# EventDB server
eventdb.server=127.0.0.1
```

Two MNC systems and a backend server on different host

This setup includes 3 different hosts:

- One central server running the backend service, event database SatDB datbase.
- Two hosts running the MNC service.

The backend service connects via network to the MNC services on both host. And the MNC services deliver their events to the central server and using the SatDB on the central server.

```
# MNC servers
backend.mnc.1.ip=10.16.2.1
backend.mnc.1.name=UPLINK-TP1
backend.mnc.2.ip=10.16.2.2
backend.mnc.2.name=UPLINK-TP2

# EventDB server
eventdb.server=10.16.2.21

# SatDB server
db.satlist.host=10.16.2.22
db.satlist.user=postgres
db.satlist.password=mysecretpassword
db.satlist.usessl=true
```

backend server controlling redundant MNC systems

This setup includes also 3 different hosts:

- One central server running the backend service, event database SatDB datbase.
- A redundant pair of MNC services (Main and Backup)

The backend service connects via network to both hosts and control which one is the currently active MNC service. The backend services monitors the availability and stops & starts the MNC services on the hosts as required.

The active MNC service delivers its events to the cental EventDB server

```
# MNC servers
backend.mnc.1.ip=10.16.2.1
backend.mnc.1.name=UPLINK-TP1
backend.mnc.1.backup.ip=10.16.2.2
backend.mnc.1.backup.retries=3
backend.mnc.1.backup.delay=30

# EventDB server
eventdb.server=10.16.2.21
```

4.2.2 user-ip-restrictions.json

If the backend finds this file at startup, it restricts the access to the backend following the rules defined in this file. The rules in the file describe which user may login to the backend from which client machine. This applies to WebUI logins, not to operation system level (ssh) logins.

IP based login restrictions be defined in two ways:

1. From a given client IP address (or subnet) only a defined set of users may log in.
2. A given user may login only from a defined set of IP addresses or subnets.

While these restrictions at the first glance sound quite similar, the consequences of these restrictions may be completly different. A type 1 rule is used to exclude all users who are not listed in the rule, a type 2 rule exludes all non listed locations for a particular user.

The user-ip-restrictions.json file uses JSON to describe the rules to be applied. Example:

```
{
   "comment": "Example rules file",
   "ip_sources": [
      {
         "source_address": "10.10.21.0/24",
         "allowed_users": ["peter","paul","mary"]
      },
      {
         "source_address": "192.168.0.144",
         "allowed_users": ["guest"]
      }
   ],
   "users": [
      {
         "username": "john",
         "allowed_addresses": [ "10.10.1.0/24", "192.168.2.0/24" ]
      }
   ],
   "default_behavior": "grant"
}
```

The array "ip_source" contains the list of type 1 rules. Each rule defines an IP address or subnet and the list of users who are permitted to login from this network location. In the example above, from the subnet "10.10.21.0/24" only the users "peter","paul" and "mary" are permitted to login. all other users are rejected when trying to login from this subnet, even if they entered valid credentials. The second rule in this list limits the login from the IP "192.168.0.144" to the user "guest".

The array "users" contains the list of type 2 rules. In this example there is only one rule define which limits the login for user "john" to the subnets "10.10.1.0/24" and "192.168.2.0/24". Login attempts for "john" will be rejected unless the IP address of his computer is in one of these subnets.

Finally, the definition "default_behavior" tells how to deal with users who are not covered by any rule above. Valid options are "grant" or "deny".

The file may contain type 1 and type 2 rules at the same time or only one of the arrays "ip_source" or "users". For example only one client machine may be limited to a certain user:

```
{
   "comment": "Example rules file",
   "ip_source": [
      {
         "source_address": "192.168.0.144",
         "allowed_users": ["guest"]
      }
   ],
   "default_behavior": "grant"
}
```

This file only limits the Browser at "192.168.0.144", only user "guest" may login from here. Other users are rejected when logging in from here, but they may login from any other source.

If the user-ip-restrictions.json file does not exist, not IP based login restrictions are applies. user-ip-restrictions.json uses strict JSON syntax, any comments in the file must be coded as definitions which are ignored by the parser (e.g. "comment":"...").

4.2.3 vlc.properties

Defines some general configuration settings of the VLC/M&C Server, some of them may be defined on the command line, too.

Timezone settings

By default the sat-nms software displays all timestamps in UTC. you may change this with the timezone parameter. IF CHANGED, THE PARAMETER MUST BE CHANGED IN BOTH, vlc.properties AND client.properties TO THE SAME VALUE !!

With a NMS system, the timezone for NMS, all VLCs and all clients MUST be UTC, the timezone parameter must not be defined !!

```
## Examples for non-UTC time zones:
## timezone=GMT+03
## timezone=GMT-10
```

SNMP exceptions

Print exceptions for SNMP communication usefull for debugging SNMP. Default is `no`. Exceptions will be logged to `.panic.log`

```
showSNMPExceptions=no
```

Catch Exceptions

Print more/all exceptions for SNMP communication usefull for debugging SNMP. Default is `no`. Exceptions will be logged to `.panic.log`

```
showCatchedExceptions=no
```

Startup debug

`yes` makes the software to log in detail what it does during startup Default is `no`. Messages will be logged to `.panic.log`

```
## startup.debug=yes
startup.debug=no
```

Procedure debug

Setting to `true` adds additional information to the output of verbose mode. It shows at the beginning of a driver cycle which procedures are marked as "to be excecuted" (dirty) with a timestamp and it show when these procedures are executed.

```
## procedures.debug=true
procedures.debug=false
```

Remember to the set verbose mode of the device to `true` if you want to see the additional information.

Example:

```
                        GET procedures 9 12 marked dirty 14:25:35.263
                                running procedure 9 14:25:35.264
tx(BCRX-1): /read?levl=?
rx(BCRX-1): levl=-77.45
 BCRX-1.power=-77.45
                                running procedure 12 14:25:35.493
tx(BCRX-1): /rmt?sact=?
rx(BCRX-1): sact=0
 BCRX-1.ssrch.active=0
```

Startup Delay

Adds a delay between the initialization of two device threads. Value in msecs, default 0.

```
startup.delay=100
```

Communication timeout *deprecated*

The communication timeout to be used for tty over ip connections in msecs, Default 2000. This setting is *deprecated* because the timeout can be configured via `Device-Window > Maintenance Page > Serial > Timeout` individually for each device.

```
tty.ip.timeout=2000
```

TCP/IP interface definitions *deprecated*

Defines interfaces to control devices via tcp/ip. This is deprecated because it can be easily defined via client `Device Setup Window`. There is no need anymore to define `ttyNx` pseudo-interfaces.

```
```

Local event database

If set to 'yes' the server starts it's own event database. This is for stand alone M&C systems.

VLC's integrated to a NMS don't use a local event database, they send the messages to the central event database on the NMS server instead. Also MNCs working with a common Event Database, don't use a local event database.

By default, the command line switch `-e` in `/home/satnms/satnms-start` will already enable a local event database if `SATNMS_TYPE="mnc"` in /etc/satnms.rc is set to `mnc`.

```
```

Duplicate events

If set to `yes` on a MNC system sending it's event messages primarily to a central event database, the message are also duplicated to a local event database. Default is `no`.

To make this work, `local.edb` must be set to `yes` as well.

Never set `duplicate.edb=yes` for a stand alone MNC system. All messages will appear twice in the event log!

```
duplicate.edb=yes
```

EventToSerial AddOn

AddOn for fault messages reports to external systems:

- satnms.addon.EventToSerial: sends text messages to serial interface
- satnms.addon.EventToMail: sends text messages to mail recipients
- satnms.addon.EventToScript: calls a user defined script on selected events

Both needs additional configuration in separate configuration files.

```
server.addon=EventToSerial
server.addon=EventToMail
server.addon=EventToScripts
```

Serial Control Port

To control the VLC/MNC remotely via a serial interface set this option to a valid port name. Don't use this port to control a device at the same time.

```
server.rcontrol.port=/dev/ttyS1
server.rcontrol.baudrate=9600
```

Network Control Port

The VLC/MNC server provides a SSL remote control port (read-write) at port 2015, a TCP remote control port at port 2016 (read-write) and a second one at port 2017 (read-only). To switch off either of these ports comment out the appropriate line.

```
server.rcontrol.ssl.read-write=yes
server.rcontrol.tcp.read-write=yes
server.rcontrol.tcp.read-only=yes
```

Build-in WebServer

Setting this true makes the built in web server on TCP port 2002 require the user to login with a valid user/password combination in order to see / modify the parameter tree.

```
http.requiresAuthentication=true
```

TimeSync with NMS server

The following entry specifies the behavior of the server when it detects that the VLC clock is more than 30 seconds off the NMS time. Three settings are possible:

- FAULT : A fault of the SYSTEM device is generated (default)
- LOG : The time difference is logged as a INFO message to the event log
- IGNORE : The time difference is ignored completely

Only required in NMS environment where no NTP service is available.

```
server.nms.clockdiff=IGNORE
```

Default Help Page

To configure the main help page set this option to the name of the html-page without any directory or file extension. The main help page is the page that appears when clicking the help button in the main screen. default: 2210

```
help.topic=4520
```

Channel database

The CDB client in the software is preconfigured to work in a VLC/NMS configuration. The CDB server is located at the NMS in this configuration, the connection to the CDB is *online* while the

VLC *online* from the NMS' point of view.

Beside this, two other configurations are possible: A MNC can run it's own CDB server as a self contained solution or multiple MNC servers can share a CDB running on one of them.

A MNC running a CDB server must set this to `true` . `false` disables the local CDB server

```
## cdb.localOnly=true
cdb.localOnly=false
```

A MNC which wants to use the CDB running on another MNC must uncomment this line with the IP address modified to the address of the M&C running the CDB server.

```
cdb.externServerAddress=192.168.2.214
```

Setting this true, enables some magic when editing channel names in the RX-LineSettings-CDB-KSC device: Channel names automatically get a signature containing the DVB-mode, the symbol rate and the RX pol appended. In the channel editor page this signature is not visible.

```
## cdb.LineSettings.magicname=true
cdb.LineSettings.magicname=false
```

If present, these entries make the channel / biss-key databases sync the local backup to the central database in the NMS in regular intervals. The time interval is defined in hours (floating point).

It is disabled if cdb.localOnly=true

```
## cdb.channels.syncInterval=24.0
## cdb.bisskey12.syncInterval=24.0
## cdb.bisskey16.syncInterval=24.0
```

settings for CSM spectrum analyzers

Switches the preamp for a LPTech LPT3000RX4 spectrum analyzer ON/OFF affects all LPT3000RX4 controlled by the M&C, default is preamp ON

```
csm.lpt3000rx4.preamp=ON
```

This setting is deprecated, it refers to the outdated CSM-Spectrum-Analyzer device. Actual spectrum analyzer drivers handle this internally

Universal SNMP agent

The following entries refer to the universal table based snmp agent listening at UDP port 2261.

Defaults are:

```
snmp.agent.enable=no
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=
```

`trapTargets` is a space separated list of IPv4 addresses, each optionally followed by a colon and a port number. for entries without port number port 162 is assumed.

Example:

```
snmp.agent.enable=yes
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=192.168.0.89:2162 127.0.0.1:2162
```

User configurable SNMP agent

The following entries refer to the user configurable snmp agent listening at UDP port 2161.

```
snmp.max.oid=1000
```

The agent gets initialized if a snmp.setup file exists. The agent limit the number of OIDs it creates from the snmp.setup file. The default limit is 1000 ans may be changed uing the parameter shown above.

UDP Heartbeat

These entries control the UDP heartbeat sent by a VLC to make the NMS recognize that the VLC is reachable in the network. To activate the heartbeat, set `udp.heartbeat.enabled=true`

The interval generally should not be changed. It is given in msecs, it should be set to one third of the timeout value set at the NMS.

```
udp.heartbeat.enabled=false
udp.heartbeat.enabled=true
udp.heartbeat.interval=5000
```

logParameterChanges

If set true, event log messages indicating parameter changes (logParameterChanges set true for a device) also contain the name and the IP address of the initiator of the parameter change. default is false for backward compatibility

```
## log.initiator=false
log.initiator=true
```

Prologix debug

Enable debug output for ProLogix GPIB to Ethernet converter. Disabled by default.

```
prologix.debug=false
```

Macro Scheduler

If set true, the macro scheduler automatically deletes outdated events. if false, these events remain in the schedule and can be viewed in the GUI. Default is true.

```
## scheduler.autodelete=true
scheduler.autodelete=false
```

Client Queue Size

This entry configures the client queue size (number of messages). Default value is 10000, the value which was set before as fixed size.

```
clientpeer.queueSize=50000
```

Port binding

By adding a line `bind.xxxx=a.b.c.d` the server socket at port `xxxx` can be bound to a specific address / interface

works fo the following port numbers

Port	Service
2000	Debug terminal
2001	Server-Client message transfer
2002	HTTP webserver (deprecated)
2003	M&C server control
2004	EventDB feed
2005	Live event log
2006	EventDB query
2007	JFTP data transfer

4.2.4 client.properties

Several aspects of the client behavior can be configured in the text based configuration file `/home/satnms/client.properties`. The following chapter descripes the different option and

related functionalities.

All chances in this file we be applied for every client which connects to the server after this configuration file was written.

Addressing users and hosts

All keys in this file may be prepended by a user name (the user name to log into the sat-nms GUI) or a host name. If the value for a key 'some.key' is searched, the software first tries 'user.some.key' where 'user' is replaced by the current user's login name. If this key was not found, it tries 'hostname.some.key' where hostname ist the client's host name as reported by the operating system. If this also fails, the as the last fall back the plain key is searched.

This search method gives the capability to overwrite some client properties for specific users or client hosts.

Examples:

```
## disable edit/take mode for client with hostname "pc1":
pc1.gui.useEditTake=false

## disable edit/take mode for sat-nms user "user1" on any client pc:
user1.gui.useEditTake=false
```

Time zone settings

By default the sat-nms software displays all timestamps in UTC. you may change this with the timezone parameter. IF CHANGED, THE PARAMETER MUST BE CHANGED IN BOTH, vlc.properties AND client.properties TO THE SAME VALUE !!

Examples for non-UTC time zones:

```
timezone=GMT+03
timezone=GMT-10
```

Exclusive login

Setting this true enables _exclusive login*, meaning that only client may be logged in at a time.

```
login.exclusive=false
login.exclusive=true
```

When trying to login on a system with enabled *exclusive login* and there is already another user logged in, the login dialog will report this with the message* `another client is logged in` *and gives you the option to *FORCE* your own login and kick out the connected user. Forcing login is only possible with an administrator account (privilege level 150)*

Primary client

Setting this to an IP address of a certain client computer gives this client instance an exclusive priority: no other client may force the log out of the primary client, but the primary client may log out any other client.

This option is meaningful only if login.exclusive=true

A typical application scenario is, that a client installed at the antenna site should be able to block a remote connected users.

```
login.primary=192.168.0.101
```

Automatically timed logout

Setting this to a value greater 0 enables automatically logout of users sat-nms client users after the given number of minutes of (user-) inactivity.

To make special setting for a given user or host add username or hostname in front of autologout parameters. By default this automatically logout is disabled (0).

Example disable autologout for user _operator* but set it to 5 minutes for all others:

```
## other users
 operator.autologout=0
 autologout=5
```

Device list windows access level

The privilege level necessary to open the device list in the toolbar in the VLCFrame / MCFrame windows. Default required privilege level to open the device list windows is 100 (operator).

```
devicelist.privilege=110
```

Config parameter access level

The privilege level necessary to change device setup parameters at the maintenance / setup window.

Default is 150, select a lower number will allow different users than normal administrators to change these settings. Recommendation is to define users with level 140.

```
configParameters.privilege=140
```

Macro scheduler access level

This defines the privilege level necessary to edit the schedule of the system macro scheduler.

```
scheduler.privilege=110
```

Screen element background color

True makes the configured background color of disabled screen elements override the system 'disabled' background. Works only with entry fields, read-only fields and with the searchable drop down list field.

```
gui.overrideBackground=true
```

Look and feel of read-only elements

True lets read-only elements look "enabled" even if the operator is not logged in. This is the way, satnms2 did show read-only fields. Default is *false*

```
gui.keepReadOnlyEnabled=true
```

Edit/Take mode

True makes the user interface use the edit/take mode for all device screens. This option change the behavior of device screens. With enabled edit/take mode commands will be not send directly if you enter a new value or select a dropdown entry. But all device windows will show 2 additional buttons : EDIT and TAKE.

The EDIT button enables all read/write parameter fields to change the values and the TAKE button send all changed parameters to the device.

By default this mode is disabled.

```
gui.useEditTake=true
```

**Event sort order*

True makes the event report and event log windows sort recent fault messages first (on top). Default is *false*.

```
gui.eventSortRecentFirst=true
```

Fault list window background color

If set *true*, in the fault list window warnings are shown with a yellow background, faults with a red background (may be changed by a toolbar button of the window)

```
gui.faultListColored=true
```

Fault list window line color

If set *true*, the fault list window sorts the lines fist by device, then by time (may be changed by a toolbar button of the window)

```
gui.faultListByDevice=true
```

Suppress warnings fault list window

If set true, the fault list window suppresses all warnings (may be changed by a toolbar button of the window)

```
gui.faultListSuppressWarnings=true
```

Suppress signal on emblem

True prevents device icons showing the 'signal on' emblem, even if the device state is recognized as 'signal on' and an image file for the emblem has been found.

```
gui.suppressSignalOnEmblem=true
```

New fault popup

If set true, each new fault makes the fault list window pop up (or come to front). applies only to the M&C client. The -o command line switch of satnms3.gui.MCFrame has the same effect.

```
gui.faultListPopUpOnNewFault=true
```

Show tree view button

True makes the M&C client window show a toolbar button to launch a tree view window, false suppresses this button. This feature is enabled by default, the tree view button is shown if this definition is missing

```
gui.showTreeViewButton=true
```

Tree view default device icon

The treeview client/window uses this device icon to build dynamically generated device screens.

```
gui.treeview.dynamicDeviceIcon=device-rectangle.gif
```

Alternative Switch icon set

Define different switch icons for the screen editor

```
gui.editor.predefinedSwitch.1=cwgswitch
gui.editor.predefinedSwitch.2=cwgswitch-mirrored
gui.editor.predefinedSwitch.3=ccxswitch
gui.editor.predefinedSwitch.4=ccxswitch-mirrored
gui.editor.predefinedSwitch.5=clswitch
gui.editor.predefinedSwitch.6=crswitch
```

VLC player integration

The full path of the VideoLAN media player. Required to run the VLC player on Windows client machines from a "VideoLAN" frame button.

Backslashes must be doubled, the whole path must be enclosed in double quotes if it contains spaces. You can also defined this setting by host to target client PCs with different operating systems.

```
videolan.exe="vlc.exe"
videolan.exe="C:\\Program Files (x86)\\VideoLAN\\VLC\\vlc.exe"
```

Event DB

Defines the IP Address of the EventDB Server used by the client to show the Eventlog. This is usually *not* configured: The Client will use the IP address of the MNC server.

In case the EventDB is not available locally on the MNC server but on any other machine you can configure here the IP address of the "external" EventDB Server.

```
eventdb.server=10.10.1.111
```

4.2.5 satnms.rc

working mode

Select between operation as MNC server (mnc) or together with an NMS Server (vlc)

```
SATNMS_TYPE="mnc"
```

serial ports

Moxa PCI Express cards did not have any jumpers to select RS232/422/485 and termination. This has to be set with `muestty` on startup.

Servers with OS Debian-satnms-4 and later supports 1 card with up to 8 ports. Define here the existing ports and interface types:

```
## add 2 lines for each port and replace "x" with the number 1..8
SATNMS_SERIALx_INT = [RS232,RS422,RS4852W,RS4854W]
SATNMS_SERIALx_TERM = [NONTERM,120TERM]
## (only relevant for 422/485)

## example Port 1 to RS422 and no termination
SATNMS_SERIAL_INT_1=RS422
SATNMS_SERIAL_TERM_1=NONTERM
```

Please refer to Moxa documentation for details. Check current settings with

```
muestty -g /dev/ttyS11
```

Card comes up with RS232, therefore it is not necessary to configure these cards. Just comment out lines with RS232 to prevent error messages.

4.2.6 system.name

Add one line (text) which should be shown in the window title of the sat-nms MNC Java Client. Default is `sat-nms MNC Monitoring & Control System`

Examples:

- Antenna 1 M&C
- MNC Teleport Steisslingen

4.2.7 traprec.json

This file configures the trap receiver built into the M&C server. The trap receiver is used to call `PROC ASYNC TRAP` procedures in selected devices / device drivers on every received SNMP trap. The trap receiver is started if this file is present and can be parsed successfully.

File Syntax

The file uses JSON syntax to define the traprec configuration in a structured way. The elements of the JSON document are explained best by example:

```
{
   "comment": "sat-nms SNMP trap receiver configuration",
   "general": {
      "listenPort": 2162,
      "logToStdOut": true,
      "logToEdb": false,
      "trapCommunity": "public"
   },
   "snmpv1": {
      "enabled": true
   },
   "snmpv2c": {
      "enabled": true
   },
   "snmpv3": {
      "enabled": true,
      "users": [
         {
            "name": "secretname",
            "authpass": "AuthPassword",
            "authentication": "MD5",
            "privacy": "NONE",
            "privpass": "PrivPassword"
         }
      ]
   }
}
```

general

The *general* section defines some general properties of the trap receiver. The definitions in this section are:

name	type	default	description
listenPort	integer	2162	The port number to listen at. The valid range for this is 1024 to 65535. If you require the trap receiver listen at the standard port 162, you should setup a port forwarding to 2162 for this.
logToStdOut	bool	false	If enabled, every received trap gets logged to stdout (to the .panic.log file). The complete trap information as available in the PROC ASYNC TRAP procedures ist logged. This is useful for debugging which traps are sent by a device.

name	type	default	description
logToEdb	bool	false	If enabled, every received trap gets logged in the M&C event log.
trapCommunity	string	null	If set, the trap receiver accepts only traps which match the given trap community (SNMPv1 and SNMPv2c only)

snmpv1

The *snmpv1* section contains a flag to enable or disable SNMPv1 at the trap receiver.

name	type	default	description
enable	bool	false	*true* enables SNMPv1

If disabled, the trapreceiver ignores all SNMPv1 traps it receives.

snmpv2c

The *snmpv2c* section contains a flag to enable or disable SNMPv2c at the trap receiver.

name	type	default	description
enable	bool	false	*true* enables SNMPv2c

If disabled, the trapreceiver ignores all SNMPv2c traps or inform messages it receives. SNMPv2c inform messages are not acknowledged by the trap receiver if SNMPV2c is disabled.

snmpv3

The *snmpv3* section contains a flag to enable or disable SNMPv3 at the trap receiver and the list of authenticaed users the trap receiver accepts.

name	type	default	description
enable	bool	false	*true* enables SNMPv3
users	array	-	a list of user definitions (see below for details)

If disabled, the trapreceiver ignores all SNMPv3 traps or inform messages it receives. SNMPv3 inform messages are not acknowledged by the trap receiver if SNMPV3 is disabled.

users

If SNMPv3 is enabled, the *snmpv3* section must contain a *users* array with at least one element. SNMPv3 does not work without a user definition, even in noAuthNoPriv mode. The definitions

for one user are:

name	type	default	description
name	string	-	The secret name of the user.
authpass	string	-	The user's pass phrase, this is required if *authentication* is not "NONE"
authentication	string	"NONE"	The authentication mode, one of "NONE", MD5" or "SHA". "NONE" implies noAuthNoPriv mode, *privacy* is ignored in this case.
privacy	string	"NONE"	The privacy mode, one of "NONE", "AES" or "DES".
privpass	string	-	The privacy pass phrase, this is required if*privacy* is not "NONE"

The *name* definition is mandatory for a user as well as the*authentication* and *privacy* definitions. The pass phrase definitions are only required if *authentication* and/or *privacy* are enabled.

4.3 Event Database Configuration and Event Notification Add-Ons

Basic EDB settings

The Event Database basic configuration (e.g. if the EDB should be started on a MNC system) is defined in `/home/satnms/vlc.properties`. Please refer to the chapter vlc.properties for details.

EDB settings in the SYSTEM device

There are also settings regarding the Event Database (e.g. IP address of the EventDB Server) which needs to be configured in the SYSTEM device

EDB.properties

Event Database properties are definesin the `/home/satnms/edb/EDB.properties` file. Page EDB.properties describes this feature in detail.

Event Notification Add-Ons

There are two add-ons which may be configured with the M&C software to notify operators if new event messages have been issued by the system. Notification may be through a RS232 serial interface of the M&C server or by E-mail. Event notification add-ons are available for sat-nms M&C systems only, not for the NMS server.

Event To Serial Notification

The event to serial notification add-on sends event messages through a serial RS232 interface.

This interface may be connected to a separate unit which sends SMS messages or teletype messages from this information. Page Event To Serial Notification describes this feature in detail.

Event To Mail Notification

The event to mail notification add-on sends event messages as e-mail notifications to one or more recipients. Page Event To Mail Notification describes this feature in detail.

Add-On Configuration

A notification add-on is activated by setting the key "server.addon" in the file "vlc.properties" either to "EventToSerial", "EventToMail" or to "EventToScript". All particular settings for the selected notification add-on are defined in a separate configuration file. A file called "rules" has to be created with the filter rules which messages are to be forwarded and which not.

The "rules" File

All event notification add-ons provide a filter function for the event messages to forward. Using filter rules defined in the file "rules", the software forwards only messages which match one of the rules specified in the "rules" file.

This file is a plain text file containing one rule per line. Empty lines and comment lines (starting with a '#'-character) are ignored. Each rule has the format <device-name>:<text-pattern>, where both parts may be replaced by "*" as a general wildcard. A rule matches, if the event source matches the given device name and if the message text contains the given text pattern.

Here some example rules:

```
HPA-1:FAULT          matches messages from HPA-1 containing
                   "FAULT" in the event message text.

HPA-2:*              matches all messages from device HPA-2

*:Lock               matches messages from all devices which
                   contain "Lock" in the message text

*:*                  matches all messages.
```

If the "rules" file is missing or empty, the software assumes a "*:*" rule. You may have to create a new "rules" file on your, system, there is no default file installed.

4.3.1 Event To Serial Notification

The event to serial notification add-on sends event messages through a serial RS232 interface. This interface may be connected to a separate unit which sends SMS messages or teletype messages from this information.

To use this feature, you need an unused RS232 port at the M&C server. Configuring / installing

this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2s.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToSerial" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2s.properties" file contains the parameters which configure the behavior of the event to serial notification. The file follows the Java properties file syntax and may contain thefollowing keys / values:

hostName specifies the hostname or IP address of the machine which runs the M&C server. In most cases you will leave this at 'localhost'.

ruleFileName specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

portName specifies the name of the serial port to use. The default value is '/dev/ttyS0', every serial port available at the machine which is not in use for another purpose may be used.

baudRate specifies the baudrate to be used with the serial port. Default is '9600'.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

4.3.2 Event To Mail Notification

The event to mail notification add-on sends event messages as e-mail notifications to one or more recipients.

To use this feature, the M&C server must be able to reach an SMTP mail server in the network. This may be a private mail server in the company LAN or a public mail server. For the latter case, the M&C server requires internet access.

Configuring / installing this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2m.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToMail" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2m.properties" file contains the parameters which configure the behavior of the event to mail notification. The file follows the Java properties file syntax and may contain thefollowing

keys / values:

hostName specifies the hostname or IP address of the machine which runs the M&C server. In most cases you will leave this at 'localhost'.

ruleFileName specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

originate specifies the name or some identification of the M&C system. The text given here will be referenced as the message source in the text body.

minSendInterval specifies the minimal time (minutes) between two messages sent by the notification add-on. For instance, setting this to '60' will configure the software to send mails not more often than once an hour. Events happening in this time will be accumulated and included to the next mail as a list.

summary specifies the position of the message summary in the mail body. It may be one of 'none', 'top' or 'bottom'. If no summary is defined, 'top' is assumed as the default.

smtpHost specifies the name or IP address of the SMTP server which shall be used to send the mails.

smtpPort specifies the port number of the SMTP server which shall be used to send the mails. If not specified, the program assumes port 25 (SMTP).

smtpUser specifies the username to login at the SMTP server. if left empty or commented out, the software uses no SMTP authentication.

smtpPassword specifies the password to login at the SMTP server.

smtpSSL may be one of yes/no. Specifies if SMTPS (SMTP via SSL/TLS) shall be used as message transport. If not specified, the program uses no encryption.

smtpFrom specifies the from address to be set in the messages sent. Please note, the many public SNMP servers require this address to be a valid address because of their spam protection policy.

smtpTo specifies the list of recipients for the mail. Multiple e-mail addresses may be separated by comma characters, they all have to appear in the same line as the 'snmpTo' key.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

4.3.3 Event To Script Notification

The event to script notification add-on calls a user defined script on each event passing the 'rules' filter.

Configuring / installing this feature is a four step procedure:

1. Create a "rules" file. See page Event Notification Add-Ons for details about this file.
2. Edit the "e2c.properties" file for your needs, some information about the settings in this file are given below.
3. Edit the "vlc.properties" file and set "server.addon=EventToScript" This line probably is already presented in the file but deactivated by a '#' character in front of it.
4. restart the M&C server

The "e2c.properties" file contains the parameters which configure the behavior of the event to script notification. The file follows the Java properties file syntax and may contain thefollowing keys / values:

scriptName - specifies the name of the (bash-) script to be executed. The file *must* be located in the ./scripts/ directory of the ***sat-nms*** installation. The file must be set readable and executable for the user that shall execute it.

ruleFileName - specifies the name of the rule file used by the notification add-on. By default this is 'rules', you may change this if you like to keep multiple rules files which you want to use alternatively.

user - if this key is defined, the software does execute the the script with `sudo -u <user> <scriptName> ...` instead of calling the script directly. You probably have to set the password property as well to make this work.

password - specifies the login password of the user set with the 'user' property.

verbose is for software tests only. For normal operation this always must be 'false'.

The file is read once at system start, hence any changes to this file become effective with the next restart of the M&C server.

4.3.4 EDB.properties

Storage depth

Depth defines the number of days the event database records in it's circular storage on disc. Typicall value is 1 year (365 days)

```
depth=365
```

Maximum Report Size

MaxReportSize defines the maximum number of events which are included in a database report. If more events match the search criteria, the reports gets truncated and the operator receives a message about this. This value is just the default used by the client event report window. There you can enter a new one (even greater) for a single request.

```
maxReportSize=1000
```

Event sort order

SortEvents controls if the filtered events are to be sorted by creation time before they are reported to the client. This is necessary in an environment where the events stored in another order than they were created (e.g. in the NMS or if multiple MNC systems using the same event database).

Therefore sortEvents is assumed to `true` if not stated otherwise. In a MNC environment reporting a large number of events can be speed up significantly by settings this to `false`

```
sortEvents=false
## sortEvents=true
```

Event DB debugging

Set to `true` for activation of additional debug output. It is disabled by default `false` and should be not enabled in production environments.

```
```

4.4 Remote Control

The satnms M&C software support several remote control interface which let an superordinate NMS control the M&C. In this section the direct remote interfaces of the M&C server are discussed. Both require direct IP access to the M&C server, accepting any possible security riscs arising from that.

The preferred method of monitoring or controlling an M&C from an external program is the HTTPS based API of the M&C backend. USing this, all M&C systems controlled by the backend may be monitored thru one single interface.

The direct remote interfaces of the M&C server areSNMP and a text based terminal protocol using TCP/SSL Ports for remote contol.

The SNMP Agent built into the M&C server comes in two different flavors: One is table based and standardized, offers a limited number of parameters in a MIB which is the same for all satnms M&C systems. The second one is application specific, for every satnms M&C installation a unique MIB can be generated which contains exactly the parameter which shal be monitored or controlled.

4.4.1 SNMP Control

The sat-nms software contains two SNMP agents which may be used to control the software from an external NMS.

Universal SNMP Agent

The universal SNMP agent publishes a limited number of device parameters in a fixed, table oriented MIB. This SNMP agent permits to monitor and partially control any sat-nms M&C using

the same MIB - regardless of the device configuration of the particular M&C. The universal SNMP agent listens at port 2261, it is based on SNMPv2.

The MIB provides one table which contains all devices which are configured in the sat-nms M&C. No special mapping is necessary for this table as there are only very common parameters like the device name, its fault state or its administrative state in this table.

Beside this, the MIB contains a couple of tables, each representing all devices belonging to a so called 'device class'. For instance, there is one table containing all HPAs in the M&C and one table containing all waveguide switches. These class specific tables contain some parameters which are common to the devices of the particular class. The HPA table contains parameters for tx-on, for attenuation and forthe measured output power for each device, the waveguide switch table provides a parameter for the switch position.

As device drivers sometimes use different parameter names for the same or similar function, an individual mapping from the 'universal' parameter in the MIB to the real parameter in the device is necessary for every single sat-nms device driver. Device drivers which do not contain any information about the device class this devices belongs to and how the parameters are to be mapped into the MIB only appear in the in the general device table of the MIB.

A sat-nms device (physical or logical) may be represented by multiple device classes in the MIB or by multiple instances of the same class. For example a combined upconverter and HPA may appera as a HPA in this table (with its HPA specific parameters) and a second time in the upconverter table with its frequency parameter.

Also devices consisting of multiple modules can be treated this way in the MIB, there are quad block upconverter on the market which provide four independent converter modules in one device. Such a device can be mapped in way, that it appears as four BUC devices in that table.

By default the universal SNMP agent is disabled. Use the following keys in the vlc.properties file to enable the agent, set the read/write community strings and an optional list of trap/notification destinations. Each IP address in this list will receive a notification message for each device which changes its fault state. Example:

```
snmp.agent.enable=yes
snmp.agent.readCommunity=public
snmp.agent.writeCommunity=private
snmp.agent.trapTargets=192.168.0.89:2162 127.0.0.1:162
```

'snmp.agent.trapTargets' defines a list of zero or more IP addresses in dotted quad notation. The addresses may contain a port number like in the example above, if no port number is specified, the standard SNMP trap port 162 is assumed. There is no fixed limitation in in the software for the number of trap destinations, you should however be aware, that a large number of destinations (greater than 10) may cause performance problems, depending on network speed and response time of the targets.

The agent sends a SNMPv2 notification (mncDeviceFaultChanged from the MIB) every time the fault state of a device changes. The notification message has the name of the device which

caused the notification and its new fault state as varbinds attached. At program start, one initial notification message for each device is sent.

Application Specific SNMP Agent

The application specific agent provided by the sat-nms software is based on SNMPv1, it accepts requests at UDP port 2161. It permits read and write access from the 'public' community. Beside that, the sat-nms M&C may be configured to send SNMP traps on changes of certain parameters, e.g. fault flags.

Each M&C system is configured individually for the device setup it controls. Therefore also the SNMP agent configuration and the specific MIB for a M&C must be unique for each installed system. With the sat-nms M&C / VLC, the SNMP agent gets configured by a (text based) configuration file which defines the M&C parameters which shall be visible through SNMP.

The typical procedure to prepare the sat-nms M&C for SNMP access is as follows:

1. Create a template SNMP configuration file using the 'create-snmp.template' utility program. This template file contains all parameters available in this particular M&C installation.
2. From this template file edit the SNMP configuration setup or simply copy the template to the configuration file if you want to export all parameters via SNMP.
3. Restart the M&C server to make it read the SNMP configuration file.
4. Invoke the 'create-satnms.mib' utility program to create a MIB definition file reflecting the actual setup.

4.4.1.1 SNMP Parameter Translations

SNMP treats parameters (objects) in a different way as the sat-nms software does. The SNMP agent translates the internal representation of M&C parameters to the SNMP representation and vice versa. The default translation rules are listed in the table below:

Default Translation

- sat-nms *data type* --- *SNMP data type**comment*
- INTEGER --- INTEGER : No translation necessary.
- FLOAT --- INTEGERSNMPv1 : cannot handle floating point numbers. FLOAT variables therefore are translated to integer values when exported as a SNMP object. The value is scaled by 10^ *n* where *n* is the number of precision digits set for this variable in the device driver definition file. Example: The azimuth value of an antenna controller appears with three precision digits as 180.000 in the sat-nms user interface. The SNMP representation of this angle will be 180000 (floating point value multiplied by 10^3). When the SNMP agent receives values to set at the M&C, the translation happens the other way round.
- TEXT --- OCTET-STRING : Text type variables translate as OCTET-STRING SNMP objects. In the MIB, text variables should be declared as 'DisplayString' (defined in RFC1213-MIB). This ensures, that the strings are shown as expected by the MIB browser. (Some browsers assume OCTET-STRINGs to be binary and display a hex-dump instead of the textual contents.)

- CHOICE --- INTEGER : Internally the sat-nms software treats CHOICE parameters as text. SNMP uses enumerated integers for this. The value of a sat-nms CHOICE parameter translates to the position of the value in the list of choices as defined in device driver file. If - for some reason - the value does not match any of the selections given by the device driver, it translates to 0. The MIB should define memnotics for all selections.
- BOOLEAN --- INTEGER : A boolean value translates to an enumerated integer with the values 0=FALSE and 1=TRUE.

The SNMP agent knows from the sat-nms variable definition if a parameter may be accessed read-write or read-only. Each individual parameter may be limited to read-only access when exported to SNMP (see below).

'force to text' Translation

The SNMP agent may be configured (individually for each parameter) to force a text-translation to OCTET-STRING, regardless of the type of the parameter. In this case, INTEGER or FLOAT values translate to a readable text, formatted with the precision defined in the device driver file and the unit string appended. BOOLEAN value translate to the keywords TRUE or FALSE and any other parameter translate the text the sat-nms user interface shows. The MIB should contain a 'DisplayString' entry for each parameter which is configured this way. If the agent receives a SET request for a parameter which ist translated in 'force to text' mode, it parses the text and sets the value as the user interface would do.

The 'force to text' option specially is valuable together with the 'force to readonly' option described below. It provides a simple method to create easy to read status reports which can be displayed in a MIB browser.

'force to readonly' Translation

The 'force to readonly' option permits to limit the access to read-only for each individual parameter. This may be used to prevent SNMP users from changing sensible settings in the M&C.

'suppress range limits' Translation

The 'suppress range limits' option suppresses the output of the min/max range of a numeric parameter. This may be useful if this range dynamically changes and shall not be fixed to the limits valid at the time when the MIB is created.

Limitations

Not all M&C parameters may be translated to SNMP objects in a way that makes sense. First of all, there are a couple of M&C variables which carry complex information, binary coded as Java objects. Translating these objects to SNMP is not possible as the information contained in these objects is for internal use in the sat-nms software only.

Another problem of translating values between their sat-nms representation and SNMP objects arises if the M&C dynamically changes the definition of this variable during runtime. This is a special feature of the sat-nms software which makes it's internal message concept superior to SNMP for this application.

Example: A satellite modem is capable to encode with FEC rates 1/2, 3/4 and 7/8 with BPSK modulation. If switched to QPSK, also the rates 4/5 and 5/6 are available. The device driver will change the definition of the 'fec' parameter as the 'modulation-type' parameter changes.

It is easy to understand that the translation of the 'fec' along the enumeration list given in the device driver will give wrong results after the device driver changed the parameter definition. It is strongly recommended to configure such parameter as 'force to text' and 'force to readonly'. This way at least a correct display via SNMP can be achieved.

A similar situation arises if a device dynamically changes the limits for a numeric parameter. An example for this are the pointing angles of an antenna: The soft limits set by the operator will be used as range limits for this parameter. If they are changed after the MIB has been created, they will no longer match the limits stated in the mib. Such parameters should be marked with the 'suppress range limits' translation option in the configuration file.

4.4.1.2 Configuring the SNMP Agent

The application specific SNMP agent gets configured by a the file 'snmp.setup'. If this file exists in the VLC's home directory while the M&C server starts, the SNMP agent is started. The sat-nms software contains a utility program called 'create-snmp.template' to create a template of this file including all parameters of a particular M&C / VLC. The file is a text file using the following format:

- Empty lines and lines starting with '#' are ignored.
- At the beginning of the file, the global settings 'systemname', 'systemid' and 'traphost' must be defined (see example below).
- After this file header, each line must contain exactly one sat-nms parameter identifier, consisting of the device name followed by a dot and the variable identifier.
- Each parameter identifier may be followed by zero or more parameter attributes, separated by space characters.

Global settings

The file 'snmp.setup' must define the global settings 'systemname', 'systemid' and 'traphost' before the first parameter identifier appears in the file. The order of appearance does not matter.

- **traphost<aaa.bbb.ccc.ddd>** --- Defines the IP address of the host which shall receive any traps from this M&C. Traps always are sent to the standard port number 162.
- **systemid <n>** --- Defines the base node number of the MIB tree created for this M&C. The created MIB tree is a branch of *iso.org.dod.internet.private.enterprises.satservice* , the 'systemid' parameter defines the number of the child node below *satservice* . Numbers 1 .. 99 are reserved for special applications. The 'create-snmp.template' utility program uses the M&C serial number as a default for this parameter to ensure a unique value for each M&C.
- **systemname <name>** --- Defines the memnotics for the 'systemid' parameter described above. Following the conventions for MIB definition, this should be a single word, starting with a letter and consisting entirely of letters and digits. The 'create-

snmp.template' utility program prepends 'vlc' to the serial number in order to create a unique value for this parameter.

Parameter attributes

The software accepts the following parameter attribute keywords following a parameter identifier. The activate the corresponding translation option for this parameter.

- **RO** --- Forces this parameter to be treated as read-only, even if it has read-write access within the sat-nms environment. It is not necessary to mark parameters with **RO** if they already are marked as read-only by the sat-nms device driver.
- **TEXT** --- Forces this parameter to be translated into a readable text (OCTET-STRING) when exported to the SNMP world, regardless of it's native data type. See chapter SNMP Parameter Translations for more information about the software converts native data types to SNMP types an vice versa.
- **NORANGE** --- Suppresses the output of a mi/max range for numeric parameters. This is useful for parameter which dynamically change their limits. See chapter SNMP Parameter Translations for more information about the software converts native data types to SNMP types an vice versa.
- **TRAP** --- Tells the software to send an SNMP trap if this parameter changes. Although mostly used for fault flags, this feature is not limited to this application. Any M&C parameter may be monitored by sending TRAPS.

MIB tree creation

When reading the 'snmp.setup' file, the software dynamically allocates MIB nodes for each parameter. The program splits up parameter identifiers at the dots and builds a MIB tree breaking down the M&C parameter structure along devices and subsystems. Node numbers are assigned automatically starting at 1, parent nodes are created as needed. This means, the order of the parameters in the 'snmp.setup' file determines the numbering of OIDs.

As a consequence of the automatic MIB tree creation, existing MIBs usually become invalid after a configuration change in a M&C, due to the renumbering of the OIDs.

However, with the knowledge about the OID assignment algorithm used by the software, minor configuration changes can be worked into the SNMP configuration while keeping the existing OID numbering unchanged. There are simply two rules to follow:

1. Do not remove existing definitions. Neither single parameters nor complete devices. Deleting definitions would cause a renumbering (decrement) of the following nodes.
2. Add additional parameter definitions always at the very end of the 'snmp.setup' file. Although for human readers the order of parameters in the file will look somewhat 'messy', the software will assign the OIDs in a way that the numbering of existing parameters will not change.

If you follow the rules above, you still can use the previous MIB of a M&C after an extension of the configuration. You however will not be able to access the added parameters as long you use the old MIB.

'snmp.setup' Example

The example below shows the contents of a short 'snmp.setup' file.

```
### snmp.setup example
### created on Thu Jun 22 19:11:18 GMT 2006

traphost 127.0.0.1
systemname vlc2345
systemid 2345

SUMMARY.fault                TRAP
E1.fault                   TRAP
E1.mode                     RO
SW.fault                   TRAP
SW.mode                     RO
```

Fault messages

If the software encounters problems while it reads the 'snmp.setup' file, it issues fault messages to the sat-nms event log. Hence, after a change of the SNMP configuration, you should inspect the event log after the server restart. The following error messages are possible:

Unknown variable in line n of 'snmp.setup': The snmp.setup file references a sat-nms variable which does not exist. The reason may be a misspelled parameter name or you changed the sat-nms device setup without adapting the SNMP configuration.

Unknown attribute ABC in line n of 'snmp.setup': You specified a parameter attribute which is neither 'TRAP', 'TEXT' nor 'RO'.

Warning: 'snmp.setup' has 0 entries: There is a 'snmp.setup' file, but it does not specify any parameters

Failed to read 'snmp.setup': A general error occurred. Delete and recreate 'snmp.setup'.

4.4.1.3 The 'create-snmp.template' Utility Program

To assist you in setting up the SNMP configuration file, the sat-nms software comes with a utility program called 'create-snmp.template'. This program creates a template configuration file which contains all (or the most common) M&C parameters of the actual system.

To create a template configuration file, perform the following steps:

1. Open a telnet window to the M&C server
2. Log in as user 'satnms', password 'satnms'.
3. Type 'create-snmp.template <ENTER>' to invoke the program

This creates a file called 'snmp.template', a template for the 'snmp.setup' file. It defines all M&C parameters of the system except of the device setup variables to be accessible via SNMP.

The M&C variables appear in alphabetical order, all fault flags are marked to send SNMP traps in case of a change of the value. To use the template file, rename it to 'snmp.setup' and restart the software. To make certain parameters invisible for the SNMP NMS, remove the corresponding lines from the file or put comment characters ('#') in front of them.

The 'create-snmp.template' utility program accepts a number of command line options which let you modify the default behavior of the program. These options are:

- **--include-config** --- Includes the device setup variables into the template file. Normally the program skips these variables.
- **--view-only** --- Forces all parameter to be read-only and translated to text (parameter attributes 'TEXT RO'). This is a fail safe mode, making it impossible for the SNMP user to change anything at the M&C system.
- **--comment** --- This option places a comment character in front of line of the file. This is valuable if you intend to export only a small number of parameters to the SNMP world. By default all parameters are disabled, you may enable the SNMP export of particular parameter by removing the comment character in front ot the parameter name.

For example, typing 'create-snmp.template --include-config --comment <ENTER>' in the telnet window creates a template file which also includes the device setup variables, but with all parameter definitions commented out.

4.4.1.4 The 'create-satnms.mib' Utility Program

Once the SNMP agent of the VLC / M&C has been configured and the server has been restarted, the agent is active and listens on UDP port 2161 for SNMP requests. At this point, calling the 'create-satnms.mib' utility program causes the server to write a MIB file which exactly matches the agent's internal MIB.

Using the the 'create-satnms.mib' utility program is as follows:

1. Open a telnet window to the M&C server
2. Log in as user 'satnms', password 'satnms'.
3. Type 'create-satnms.mib <ENTER>' to invoke the program

This creates the MIB file (file name 'satnms.mib') in the server's home directory. The file also appears in the 'public' share as a symbolic link, making it readable to other computers in the network.

4.4.2 TCP/SSL Remote Control

sat-nms M&C systems can be controlled remotely using a simple text based IP protocol. A remote application may use this protocol to query state information or device settings and to control the settings of each individual device. The M&C provides three flavours of this protocol on three different IP ports:

- Port 2015 accepts SSL connections giving monitor and control access to all parameters / devices in the M&C
- Port 2016 accepts plain TCP connections giving monitor and control access to all

parameters / devices in the M&C
- Port 2017 accepts plain TCP connections like port 2016, but grants monitoring access only.

All three ports are disabled by default. In order to use them, they have to be enabled in the vlc.properties file:

```
server.rcontrol.ssl.read-write=yes
server.rcontrol.tcp.read-write=yes
server.rcontrol.tcp.read-only=yes
```

To make changes in the vlc.properties file effective, you must restart the M&C server. The protocol itself is very simple, there are only three commands:

```
? *parameter-id*
```

Queries the actual state of a parameter.*parameter-id* is a textual string as used in the configuration of logical devices, e.g. BCRX-1.power identifies the power reading of a SatService beacon receiver named BCRX-1. The parameter may be any device setting or a status reading. If there is a value known for the given parameter-id, the parameter-id followed by a space character and the actual value is replied, numeric values are formatted in the same way as for presenting them at the sat-nms GUI. If the parameter-id is not known or if the device driver could read a value from the device since start of the program, nothing is replied.

```
@ *parameter-id*
```

Subscribes for updates on every change of a parameter. If there is a value known for the given parameter-id, the parameter-id followed by a space character and the actual value is replied, numeric values are formatted in the same way as for presenting them at the sat-nms GUI. If the parameter-id is not known or if the device driver could read a value from the device since start of the program, nothing is replied. Every time the value of this parameter changes, the M&C will send a line with the parameter-id followed by a space character and the actual value of the parameter.

```
! *parameter-id new-value*
```

Sets a parameter to the given value. *new-value* is parsed as if entered to a field in the sat-nms GUI. There is no reply to this command. If you want to check if the new value has been accepted, use the '?' or '@' commands. At the monitoring only port 2017 set commands are silently ignored.

Remarks

- Subscriptions remain active until the connection is closed, there is no 'unsubscribe' command.
- If you are using subscriptions, it is always possible that you receive a change message for a changed parameter before you receive the expected reply to a subscribe or query

command. You always must be capable to receive messages in any sequence.

5 Installation

This section of the manual provides installation instruction for the different modules of the ***sat-nms*** Monitoring & Control software.

> ***Remark:*** On a preconfigured system delivered by SatService these step as usually **not** required. So this manual is aimed especially at people who want to deploy the system in virtual machines or customer provided bare metal server or for deployments in private cloud environment. And it is intended for a better understanding of your installation.

Quick access links to the main sections of the document:

- Base Debian Linux setup
  - packages: satnms4-base, satnms4-utils
- MNC service
  - packages: satnms4-mnc, satnms4-images, satnms4-documentation, satnms4-jars
- EventDB service
  - packages: satnms4-mnc, satnms4-images, satnms4-documentation, satnms4-jars
- Backend service
  - packages: satnms4-backend
- WebClient and nginx
  - packages: satnms4-webclient, satnms4-images, satnms4-documentation
- Database SatDB
  - packages: satnms4-satdb

There are different way where to installed all these components. The simplest solution is to install all components on one host (bare metal server or virtual machine). On the other hand all components can be run on different hosts for several reasons like redundancy or load distribution.

Please refer to the chapter Architecture for the different options. TODO!!!

The following chapters consider all components as a stand-alone installation. Therefore some steps may be repeated and can be skipped in different scenarios. But the installation of all services (MNC, backend, EventDB, WebClient and SatDB) expect the base Debian Linux installation. So whatever you want to install, make sure you followed first the instruction in the chapter Base Debian Linux setup

5.1 Installation Base Debian Linux

This section provides installation instructions for the ***sat-nms*** base and utils packages for the ***sat-nms*** Monitoring & Control System.

These packages are used to add settings and tools to a standard minimal Debian 12 installation. The following chapter descripes the installation on Debian Linux 12 but should be similar for other Debian-based distributions.

5.1.1 Requirements

- Minimal Debian 12 Installation and console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- access to Debian 12 repositories either via Internet or from a local Debian archive mirror
  - check with `apt-get update` access to the configured repository is available.
- for base setup: `satnms4-base_[version].deb` package (in the following referred to as `satnms4-base`)
- for utilities setup: `satnms4-utils_[version].deb` package (in the following referred to as `satnms4-utils`)

5.1.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

Install Debian Linux 12 on your host with your favorite installation method.

We recommend the following settings:

- Language: `English`
- Disk: 20.0 GB, ext4 filesystem
- TaskSel: not a full installation, just select
  - SSH Server
  - Standard system utilities
- Time: `UTC`
- Default user: `satnms`

Install ***sat-nms*** base components

```
apt-get update
apt-get upgrade
apt install /home/public/satnms4-base_[version].deb
apt install /home/public/satnms4-utils_[version].deb
reboot
```

5.1.3 Installation

For the following instructions we assume that you are using Debian 12. Similar Linux distributions like Ubuntu may also work but are not covered by this manual.

Debian Linux 12

Please follow the standard Debian 12 instruction for a minimal Debian installation. We

recommend the following setting for the Debian installer:

- Language: `English`
- locale: `en_GB.UTF-8`
- Network: fixed IP address
- Clock: use NTP
- Root user: set password and allow login
- Normal user: `satnms`
- Timezone: `UTC`
- Disk: 20GB partion for all with ext4
- TaskSel: only `SSH Server` and `Standard system utilities`

Additional packages

For convenience we recommend the following additional packages:

- curl
- debsums
- dialog
- htop
- mc
- rsync
- snmp
- sudo
- tcpdump
- tnftp
- vim

```
root@satnms7-dev:~# apt-get install curl debsums dialog htop mc rsync snmp sudo tcpdump
```

satnms4-base package

The `satnms4-base` package contains files and directories required by the other satnms4 components. It should be installed on each host which run ***sat-nms*** MNC components.

```
root@satnms7-dev:~# apt-get install ./satnms4-base_1.2.2-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-base' instead of './satnms4-base_1.2.2-1_all.deb'
The following NEW packages will be installed:
  satnms4-base
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/462 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /root/satnms4-base_1.2.2-1_all.deb satnms4-base all 1.2.2-1 [462 kB]
Selecting previously unselected package satnms4-base.
(Reading database ... 36810 files and directories currently installed.)
Preparing to unpack .../satnms4-base_1.2.2-1_all.deb ...
Unpacking satnms4-base (1.2.2-1) ...
Setting up satnms4-base (1.2.2-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/satnms.service → /etc/systemd/
Processing triggers for mailcap (3.70+nmu1) ...
```

Beside installing different files it will also setup the following:

- create a user `satnms` if not already exists
- create a directory `/home/public` which should be used later to place satnms4 packages and backups
- enable `sudo` for a list of commands
- add additional MOTD (message of the day) scripts to show information on login
- and some more

  Remark: All satnms4 packages must be installed from local source. All following instructions assume that you have these packages in the folder `/home/public` on the host where you want to install them.

If the user `satnms` does not exists before installing this package, a new user will be created. This user will be used to run the different ***sat-nms*** services. So it is required. By default it is not possible to login to this user, because there is no password set. Login as `root` and change the password for the user `satnms`:

```
root@satnms7-dev:~# passwd satnms
New password:
Retype new password:
passwd: password updated successfully
```

satnms4-utils package

The `satnms4-utils` package contains a collection of utilities for the***sat-nms*** Monitoring & Control System like backup and restore scripts.

```
root@satnms7-dev:/home/public# apt install ./satnms4-utils_1.0.6-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-utils' instead of './satnms4-utils_1.0.6-1_all.deb'
The following NEW packages will be installed:
  satnms4-utils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/30.7 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-utils_1.0.6-1_all.deb satnms4-utils all 1.0.6-1 [30.7 kB]
Selecting previously unselected package satnms4-utils.
(Reading database ... 36932 files and directories currently installed.)
Preparing to unpack .../satnms4-utils_1.0.6-1_all.deb ...
Unpacking satnms4-utils (1.0.6-1) ...
Setting up satnms4-utils (1.0.6-1) ...
Processing triggers for mailcap (3.70+nmu1) ...
```

Common Linux settings

In preinstalled systems delivered by SatService a dedicated file will display the version of the ***sat-nms*** Linux Operation System installation: `/etc/satnms_version` .

Its format is defined as: `[linux-distribution]-satnms-[version] [date]` . For customer build OS version it is required to create this file by your own.

- `linux-distribution` is usually `debian` , but other names are allowed like `ubuntu`
- the second word needs to be `-satnms-`
  - required because several tools parsing this file to get the OS version
  - the Sysinfo device uses this information to display the matching server information
- version format is `[major-version].[minor-version][suffix] [date]` e.g. `7.01`
  - major-number must be 7
  - minor-number will be increased if SatService creates new OS images
  - for customer builds minor numbers needs to be managed by customer
  - optional suffix can be added to mark special editions: e.g. `7.01idu` shows that it is a special OS image for the ***sat-nms*** ACU Indoor Unit
  - the date should be in format `YYYY-MM-DD`

```
satnms@satnms7-dev:~$ cat /etc/satnms_version
debian-satnms-7.01 2023-10-05
```

The follwing additional settings are recommended for convenience.

- `/home/satnms/.bashrc` : uncomment line `force_color_prompt=yes` to enable colored prompt
- `/root/.bashrc` : configure prompt colors, add the following line:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u@\[\033[01;33m\]\h\[\033[00m\]:
```

5.2 Installation MNC Service

This section provides installation instructions for the MNC service used by the***sat-nms*** Monitoring & Control System.

The MNC service runs device drivers and logical device and is the interface to the equipment to be monitored and controlled.

The following topics will be covered by the Administration Manual

- Update ***sat-nms*** MNC service TODO LINK !!!
- Backup and restore ***sat-nms*** MNC service settings TODO LINK !!!

5.2.1 Requirements

- Base Debian 12 Installation with installed packages
  - `satnms4-base`
  - `satnms4-utils`
- console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- for MNC installation the following packages:
  - `satnms4-documentation_[version].deb` package (in the following referred to as `satnms4-documentation`)
  - `satnms4-images_[version].deb` package (in the following referred to as `satnms4-images`)
  - `satnms4-jars_[version].deb` package (in the following referred to as `satnms4-jars`)
  - `satnms4-mnc_[version].deb` package (in the following referred to as `satnms4-mnc`)

5.2.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install openjdk-17-jre-headless
apt install /home/public/satnms4-documentation_[version].deb
apt install /home/public/satnms4-images_[version].deb
apt install /home/public/satnms4-jars_[version].deb
apt install /home/public/satnms4-mnc_[version].deb
systemctl enable satnms
systemctl start satnms
```

5.2.3 Installation

One external tools must be installed:

1. OpenJDK Java Runtime Environment (JRE)
2. JQ Pretty print of JSON data on CLI (optional)

Debian provided installation packages (`.deb`) for OpenJDK 17.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... ca-certificates-java java-common ...
Suggested packages:
  ...
The following NEW packages will be installed:
  ... ca-certificates-java java-common openjdk-17-jre-headless ...
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.
Need to get 50.3 MB of archives.
After this operation, 210 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
... downloading and installing packages ...
```

To test if Java Runtime was installed successfully, run `java -version` as user `satnms` :

example output

```
satnms@satnms7-dev:/home/public$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Debian-1deb12u1, mixed mode, sharing)
```

Now install the ***sat-nms*** backend software package:

- upload `satnms4-documentation` , `satnms4-images` , `satnms4-jars` and `satnms4-mnc` package to `/home/public on the target system
- login as `root`
- install the local satnms4-* packages
  - `apt install /home/public/satnms4-documentation_[version].deb`
  - `apt install /home/public/satnms4-images_[version].deb`
  - `apt install /home/public/satnms4-jars_[version].deb`
  - `apt install /home/public/satnms4-mnc_[version].deb`

example output for package satnms4-mnc

```
root@satnms7-dev:/home/public# apt-get install ./satnms4-mnc_3.68.9-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-mnc' instead of './satnms4-mnc_3.68.9-1_all.deb'
The following packages will be upgraded:
  satnms4-mnc
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/10.2 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-mnc_3.68.9-1_all.deb satnms4-mnc all 3.68.9-1 [10.2 MB]
Reading changelogs... Done
(Reading database ... 48696 files and directories currently installed.)
Preparing to unpack .../satnms4-mnc_3.68.9-1_all.deb ...
Unpacking satnms4-mnc (3.68.9-1) over (3.68.2-1) ...
Setting up satnms4-mnc (3.68.9-1) ...
```

On success you will find a multiple new files and directories `/home/satnms/` like `satnms.jar` and directories `drivers` , `protocols` etc.

5.2.4 Configuration

This section describes only the intital configuration step for the***sat-nms*** MNC service. Because the configuration of the M&C itself can be done via Web interface. Please refere to the following chapters:

- web client: Device-Setup
- web client: Screen-Editor

The system ships a default configuration which is "ready-to-run" but you can modify some configuration to adapt the system to your environment. For a all-in-one MNC System no additional configuration is required.

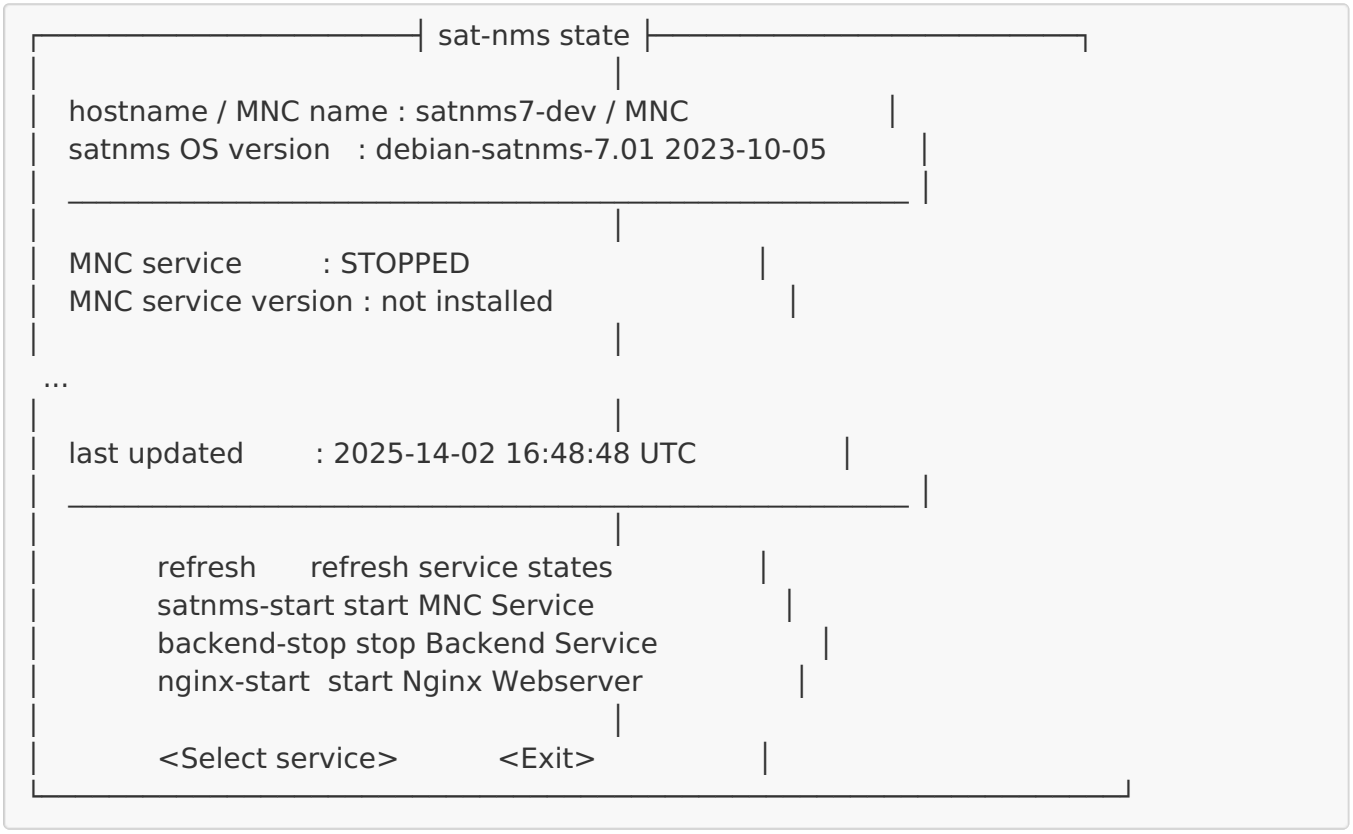
Several other options require your attention only for complex setups which includes server redundancy. A detailed description of all options from is available in the Administration Section of this Manual.

To ensure the MNC service will automatically start on reboot, login as root and enable this service.

- login as `root`
- enable autostart with `systemctl enable satnms"

Now you can start the backend service as user `satnms` and check if its running.

- login as `satnms`
- run the `satnms-watch`



The output depends on the installed services. In the example above, only the ***sat-nms*** MNC service is installed and running.

You can also control the MNC service directly with the follwing commands:

- start MNC service `sudo systemctl start satnms`
- stop MNC service `sudo systemctl stop satnms`

Check if the MNC service start successfully:

- status of MNC service `systemctl status satnms`

```
satnms@satnms7-dev:~/backend$ systemctl status backend
TODO
```

You can test if the MNC started successfully by checking the output of the log file `/home/satnms/debug.log` . The first line shows an early stage of start process and is an indication that the MNC service begins to start. The second line shows that all sub processes are now initialized and the MNC service is ready to serve client requests. Starting from this the backend service or clients are able to connect to the MNC service.

```
satnms@satnms7-dev:~$ tail /home/satnms/debug.log
...
000000 2025-03-05 12:32:18 2025-03-05 12:32:18   I SYSTEM Loading setup data (sat-nms MI
...
000000 2025-03-05 12:32:18 2025-03-05 12:32:18   I SYSTEM VLC 3.68.9 2025-03-05 07:22:0
...
```

Some more details and extended error messages if something failed are available in `/home/satnms/.panic.log`

A typical output of the default installation should looks like this:

```
satnms@satnms:~$ cat .panic.log
satnms version 3.68.9 2025-03-05 07:22:02
TRAPREC: File traprcv.json not found
TRAPREC: Cannot parse traprcv.json
TRAPREC: Trap receiver not started.
Universal SNMP agent listening at port 2261
TCP remote control server (r/w) listening at port 2016
TCP remote control server (r/o) listening at port 2017
SSL remote control server (r/w) listening at port 2015
```

A detailed description how to run the ***sat-nms*** MNC service is available in the administration manual.

Troubleshooting

If you have trouble to bring up the MNC service check first if the MNC service is still running with:

- `systemctl status satnms` or
- `/home/satnms/satnms-stat`

Additionally you can lookup the process ID (PID) of the backend service process:

```
satnms@satnms7-dev:~$  ps aux |grep satnms.jar
satnms   8204 ?   Sl   25:18 java -cp satnms.jar:/home/satnms/jars/* -mx512m -Djava.security.
```

Now check if the MNC process with this PID (here 8204 from the second column) is associated to a process listening on TCP ports from 2000 and above. The real number of listing ports depends on the actual configuration, but at least 2000-2007 should be visible.

```
satnms@satnms7-dev:~$ ss -tanp |grep 8204
LISTEN 0      50                 *:2000             *:*    users:(("java",pid=8204,fd=11))
LISTEN 0      50                 *:2001             *:*    users:(("java",pid=8204,fd=35))
LISTEN 0      50                 *:2002             *:*    users:(("java",pid=8204,fd=42))
LISTEN 0      50                 *:2003             *:*    users:(("java",pid=8204,fd=40))
LISTEN 0      50                 *:2004             *:*    users:(("java",pid=8204,fd=12))
LISTEN 0      50                 *:2005             *:*    users:(("java",pid=8204,fd=13))
LISTEN 0      50                 *:2006             *:*    users:(("java",pid=8204,fd=19))
LISTEN 0      50                 *:2007             *:*    users:(("java",pid=8204,fd=17))
...
```

!!!! The backend internal web server runs on Port 8443 and serves only HTTPS requests. ???? You should that this process is `LISTEN` on all interfaces (`*:8443`)

5.3 Installation Backend Service

This section provides installation instructions for the backend service used by the***sat-nms*** Monitoring & Control System.

The backend provides an unified REST-API to control all parts of an sat-nms Monitoring & Control System. It is used by the sat-nms WebClient but also available for third party applications. Please refere to the [API documentation](API Reference) for details.

The following topics will be covered by the Administration Manual

- Update ***sat-nms*** backend TODO LINK !!!
- Backup and restore ***sat-nms*** backend settings TODO LINK !!!

5.3.1 Requirements

- Base Debian 12 Installation with installed packages
  - `satnms4-base`
  - `satnms4-utils`
- console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission

- for backend installation: `satnms4-backend_[version].deb` package (in the following referred to as `satnms4-backend`)

5.3.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install openjdk-17-jre-headless
apt install /home/public/satnms4-backend_[version].deb
cp /home/satnms/backend/backend.properties.default /home/satnms/backend/backend.proper
systemctl enable backend
systemctl start backend
```

5.3.3 Installation

One external tools must be installed:

1. OpenJDK Java Runtime Environment (JRE)
2. JQ Pretty print of JSON data on CLI (optional)

Debian provided installation packages (`.deb`) for OpenJDK 17.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... ca-certificates-java java-common ...
Suggested packages:
  ...
The following NEW packages will be installed:
  ... ca-certificates-java java-common openjdk-17-jre-headless ...
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.
Need to get 50.3 MB of archives.
After this operation, 210 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
... downloading and installing packages ...
```

To test if Java Runtime was installed successfully, run `java -version` as user `satnms` :

example output

```
satnms@satnms7-dev:/home/public$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Debian-1deb12u1, mixed mode, sharing)
```

Optionally install JQ an lightweight and flexible command-line JSON processor for test the backend API.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install JRE from Debian repository: `apt-get install openjdk-17-jre-headless`

example output

```
root@satnms7-dev:/home/satnms# apt-get install jq
...
The following NEW packages will be installed:
  jq libjq1 libonig5
...
```

Now install the ***sat-nms*** backend software package:

- upload `satnms4-backend` package to `/home/public on the target system
- login as `root`
- install the local satnms4-backend package

`apt install /home/public/satnms4-backend_[version].deb` .

example output

```
root@satnms7-dev:/home/public# apt install ./satnms4-backend_1.17.10-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-backend' instead of './satnms4-backend_1.17.10-1_all.deb'
The following NEW packages will be installed:
  satnms4-backend
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 0 B/24.7 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-backend_1.17.10-1_all.deb satnms4-backend all 1.17.10-1 [24.7 I
Selecting previously unselected package satnms4-backend.
(Reading database ... 38210 files and directories currently installed.)
Preparing to unpack .../satnms4-backend_1.17.10-1_all.deb ...
Unpacking satnms4-backend (1.17.10-1) ...
Setting up satnms4-backend (1.17.10-1) ...
```

On success you will find a new directory `/home/satnms/backend/` which contains the software and configuration files.

5.3.4 Configuration

A ***sat-nms*** backend service provides a gate to all ***sat-nms*** MNC servers in your overall system. This could be just a single MNC service running on the same host as the backend (all-in-one server) or any number of MNC service distributed over your network.

There are three steps to configure the ***sat-nms*** backend service:

1. create configuration file `/home/satnms/backend/backend.properties`
2. modify configuration file
3. enable autostart of backend service

By default the `satnms4-backend` package delivers a sample configuration for an all-in-one server installation. This template can be found in `/home/satnms/backend/backend.properties.default` . You can use this template by copy it to real backend configuration file: `/home/satnms/backend/backend.properties` and then editing it regarding you requirements.

To enable a minimal backend setup with a MNC service running on localhost just use the provide template:

```
satnms@satnms7-dev:~/backend$ cp backend.properties.default backend.properties
```

A detailed description of all options from `backend.properties` is available in the administration manual.

To ensure the backend service will automatically start on reboot, login as root and enable this service.

- login as `root`
- enable autostart with `systemctl enable backend"

Now you can start the backend service as user `satnms` and check if its running.

- login as `satnms`
- run the `satnms-watch`

```
┌──────────────────────────┤ sat-nms state ├──────────────────────────┐
│                                                                     │
│  hostname / MNC name : satnms7-dev / MNC                            │
│  satnms OS version   : debian-satnms-7.01 2023-10-05                │
│  ──────────────────────────────────────────────────────────────     │
│                                                                     │
 ...
│                                                                     │
│  backend service     : RUNNING                                      │
│  backend version     : 1.17.10                                      │

...
│                                                                     │
│  last updated        : 2025-14-02 16:48:48 UTC                      │
│  ──────────────────────────────────────────────────────────────     │
│                                                                     │
│          refresh      refresh service states                        │
│          satnms-start start MNC Service                             │
│          backend-stop stop Backend Service                          │
│          nginx-start  start Nginx Webserver                         │
│                                                                     │
│          <Select service>          <Exit>                           │
└─────────────────────────────────────────────────────────────────────┘
```

The output depends on the installed services. In the example above, only the ***sat-nms*** backend service is installed and running.

You can also control the backend service directly with the follwing commands:

- start backend service `sudo systemctl start backend`
- stop backend service `sudo systemctl stop backend`

Check if the backend service start successfully:

- status of backend service `systemctl status backend`

```
satnms@satnms7-dev:~/backend$ systemctl status backend
● backend.service - sat-nms MNC Backend
     Loaded: loaded (/etc/systemd/system/backend.service; disabled; preset: enabled)
     Active: active (running) since Fri 2025-02-14 16:06:10 UTC; 5min ago
   Main PID: 18136 (java)
      Tasks: 22 (limit: 2307)
     Memory: 136.9M
        CPU: 8.788s
     CGroup: /system.slice/backend.service
             └─18136 java -Djava.security.egd=file:/dev/./urandom -jar backend.jar
```

You can test if the backend is ready to reply on REST-API requests use the heathcheck API-Endpoint which does not require authentication. It requires `jq` and `curl` installed. Run: `curl -k https://localhost:8443/healthcheck | jq` and if should report some information about the running backend software.

```
satnms@satnms7-dev:~$ curl -k -s  https://localhost:8443/healthcheck | jq
{
  "dict": [
    {
      "key": "PRODUCT",
      "val": "satnms4 backend"
    },
    {
      "key": "WEBSITE",
      "val": "www.satnms.com"
    },
    {
      "key": "VERSION",
      "val": "1.17.10 2024-12-17 12:22:54"
    },
    {
      "key": "SERIALNO",
      "val": "000001"
    }
  ]
}
```

Logfiles are available:

- `cat /home/satnms/backend/backend.log`
- `journalctl -u backend`

A detailed description how to run the ***sat-nms*** backend service is available in the administration manual.

Troubleshooting

If you have trouble to connect to the backend server's API endpoints check first if the backend service is still running with:

- `systemctl status backend` or
- `/home/satnms/backend-stat`

Additionally you can lookup the process ID (PID) of the backend service process:

```
satnms@satnms7-dev:~$ ps axu | grep backend.jar
satnms   18136  0.3  8.4 2606224 170268 ?   Ssl  16:06   0:25 java -Djava.security.egd=file:/d
```

Now check if the backend process with this PID (here 18136 from the second column) is associated to a process listening on the TCP port 8443:

```
satnms@satnms7-dev:~$ ss -taunp |grep 8443
tcp   LISTEN 0      100    *:8443     *:*    users:(("java",pid=18136,fd=10))
```

The backend internal web server runs on Port 8443 and serves only HTTPS requests. You should that this process is `LISTEN` on all interfaces (`*:8443`)

5.4 Installation SatDB database

This section provides installation instructions for the PostgreSQL database used by the***sat-nms*** Monitoring & Control System.

PostgreSQL is a free and open-source relational database management system. Installation packages are available for different operating systems and Linux distributions. The following chapter descripes the installation on Debian Linux 12 with the PostgreSQL version 15.

Details about PostgreSQL: https://www.postgresql.org/

If you have your own PostgreSQL installation or PostgreSQL cluster running on a different host or as a cloud service, you can skip the chapter Installation and move directly to chapter Configuration.

To update the database schema, we are using the Golang-Migrate tool. Details can be found here: https://github.com/golang-migrate/migrate.

Quick access links to the main sections of this installation guide:

- Database Installation
- Add database user
- Create SatDB database

The following topics will be covered by the Administration Manual

- Update SatDB database TODO LINK !!!
- Backup and restore SatDB database TODO LINK !!!

5.4.1 Requirements

- Minimal Debian 12 Installation and console or SSH access
- headless installation is sufficient
- root permission in Linux either via `su` or `sudo`
  - all examples in this manual assume that you get root shell with the `su` command
  - for installation with no direct `root` login, you have to add `sudo` in front of all commands which requires `root` permission
- access to Debian 12 repositories either via Internet or from a local Debian archive mirror
  - check with `apt-get update` access to the configured repository is available.
- for SatDB database setup: `satnms4-satdb_[version].deb` package (in the following referred to as `satnms4-satdb`)

5.4.2 For the impatient

This section provides a quick installation guide for the impatient and experienced Debian Linux users. Detailed information are in the following chapters.

```
apt-get update
apt-get upgrade
apt-get install postgresql
apt install /home/public/migrate.linux-amd64.deb
apt install /home/public/satnms4-satdb_[version].deb
```

5.4.3 Installation

Two external tools must be installed:

1. PostgreSQL
2. Golang-Migrate

Debian provided installation packages (`.deb`) for PostgreSQL.

- login as `root`
- update all installed Debian packages: `apt-get update; apt-get upgrade`
- install PostgreSQL from Debian repository: `apt-get install postgresql`

example output

```
satnms@satnms7-dev:~$ su -
Password:
root@satnms7-dev:~# apt-get update
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done

root@satnms7-dev:~# apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

root@satnms7-dev:~# apt-get install postgresql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ... postgresql-client-15 postgresql-client-common postgresql-common ...
The following NEW packages will be installed:
  ... postgresql-client-15 postgresql-client-common postgresql-common ...
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 49.1 MB/49.1 MB of archives.
After this operation, 200 MB of additional disk space will be used.
Do you want to continue? [Y/n]
... downloading and installing packages ...
```

By default Linux user `postgres` has access to the database from `localhost` only without password.

To test if the PostgreSQL database was installed and started successfully, you can connect with command line tool `psql` (in short CLI):

- login as `root`
- switch to the PostgreSQL user: `su - postgres`
- run the PostgreSQL interactive terminal: `psql`
    - installed version will be show
    - an interactive prompt is available
- leave PostgreSQL terminal with `\q`
- switch back to user root: `exit`

example output

```
root@satnms7-dev:~# su - postgres
postgres@satnms7-dev:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# \q
postgres@satnms7-dev:~$
```

To install Golang-migrate you need to manually download the Debian AMD64 package

- download debian package `migrate.linux-amd64.deb` from https://github.com/golang-migrate/migrate/releases/
- upload to target system in `/home/public`
- install the package with `apt-get install /home/public/migrate.linux-amd64.deb`

example output

```
root@satnms7-dev:~# apt-get install /home/public/migrate.linux-amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'migrate' instead of '/home/public/migrate.linux-amd64.deb'
The following NEW packages will be installed:
  migrate
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/16.2 MB of archives.
After this operation, 53.2 MB of additional disk space will be used.
Get:1 /home/public/migrate.linux-amd64.deb migrate amd64 4.18.1 [16.2 MB]
Selecting previously unselected package migrate.
(Reading database ... 55809 files and directories currently installed.)
Preparing to unpack .../public/migrate.linux-amd64.deb ...
Unpacking migrate (4.18.1) ...
Setting up migrate (4.18.1) ...
```

You can test the successfull installation with `migrate --version` which should display the installed version number, e.g. 4.18.1

5.4.4 Configuration

If PostgreSQL is installed you need to setup an user and a database for the ***sat-nms*** MNC system. These setups will be automatically performed if you install the `satnms4-satdb` package . Installation instructions for system where you are not able to install our Debian package, manual instructions are included in the following chapters.

The ***sat-nms*** MNC system will use the user `satnms` to connect to the database. This user will also be the owner of the SatDB. This prevents the application from requiring administrative rights for the entire database and the MNC system can use the same PostgreSQL database

engine together with other software.

The ***sat-nms*** MNC system will use a database with the name `satnms`. If you are using you own PostgreSQL installation, make sure that this database name is not already used by any other application.

- upload the `satnms4-satdb` to /home/public on the target system running PostgreSQL.
- login as `root`
- install the local satnms4-satdb package `apt install /home/public/satnms4-satdb_[version].deb`. During installation you will be prompted for a password. The default password in preinstalled systems delivered by SatService is `satnms`. Please change it to secure your system.

example output

```
root@satnms7-dev:~# apt install /home/public/satnms4-satdb_0.2.0-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'satnms4-satdb' instead of '/home/public/satnms4-satdb_0.2.0-1_all.deb'
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  satnms4-satdb
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 0 B/5,304 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/public/satnms4-satdb_0.2.0-1_all.deb satnms4-satdb all 0.2.0-1 [5,304 B]
Selecting previously unselected package satnms4-satdb.
(Reading database ... 159218 files and directories currently installed.)
Preparing to unpack .../satnms4-satdb_0.2.0-1_all.deb ...
checking user 'satnms' ...
checking postgresql ...
/var/run/postgresql:5432 - accepting connections
checking migrate ...
check database user 'satnms' ...
create PostgreSQL user 'satnms' ...
Enter password for new role:
Enter it again:
check database 'satnms' ...
Unpacking satnms4-satdb (0.2.0-1) ...
Setting up satnms4-satdb (0.2.0-1) ...
installing SatDB database schema ...
1/u init_satdb (26.36938ms)
2/u init_satdb (29.867137ms)
```

Remark: the default password in preinstalled systems delivered by SatService is `satnms` . Please change it to secure your system.

5.4.5 PostgreSQL network configuration

By default, access to the PostgreSQL database engine is restricted to the local system (the system where the database is installed) and no remote access via network is possible. If the ***sat-nms*** MNC system is a all-in-one installation this default configuration is sufficient and you can skip this chapter.

For a distributed MNC installation network access from a different host must be possible. Usually the database runs on the central server and the connected MNC servers needs to access the database via network.

- login as `root`
- edit the file `/etc/postgresql/15/main/pg_hba.conf`
  - if a different version of PostgreSQL is installed replace `15` with the version number
  - add a line at the end of the file with `host all all [local-net] md5`
    - where [local-net] is the network from where you want to connect e.g `192.168.2.0/24`
- edit file `/etc/postgresql/15/main/postgresql.conf`
  - uncomment and change line to: `listen_addresses = '*'`
- restart PostgreSQL: `systemctl restart postgresql`

example from pg_hba.conf

```
# allow access for two additional networks
host    all             all             172.17.1.0/16           md5
host    all             all             192.168.1.0/24          md5
```

example from postgresql.conf

```
# - Connection Settings -
listen_addresses = '*'   # what IP address(es) to listen on;
port = 5432              # (change requires restart)
```

You can test the network configuration with from any other system with an installed `psql` tool, or even on the local system if you specify the IP address of your PostgreSQL server:

- login as user `satnms`
- test connection: `psql --host [server-ip-address]`

example output

```
satnms@satnms7-dev:/etc/postgresql/15/main$ psql --host 192.168.2.231
Password for user satnms:
psql (15.10 (Debian 15.10-0+deb12u1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.
satnms=>
```

5.4.5.1 Change users password

To avoid that the clear text password appears in the CLI history, you can use the `\password` command in `psql`.

- login as `root`
- switch to user the PostgreSQL user: `su - postgres`
- run `psql`
- enter command `\password satnms` which prompts you for a new password

example output

```
postgres@satnms7-dev:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# \password satnms
Enter new password for user "satnms":
Enter it again:
postgres=# \q
```

5.4.6 Manual configuration

You can skip this section if you successfully installed the `satnms4-satdb` package.

5.4.6.1 Manual add database user

New users can be created via CLI or with SQL statements. This manual uses the CLI tool. There is no difference between database actions or CLI command, `createuser` is just a wrapper around the SQL command `CREATE ROLE`.

- Roles subsumes users and groups, so to be precises we are create a new role and not a user
- CLI https://www.postgresql.org/docs/current/app-createuser.html
- login as `root`
- switch to user the PostgreSQL user: `su - postgres`
- run `createuser --interactive --pwprompt` to create a new user

- name: `satnms`
- password: [set your own password here]
- be superuser: `no`
- allow to create databases: `no`
- allow to create more new roles : `no`

example output

```
root@satnms7-dev:~# su - postgres
postgres@satnms7-dev:~$ pwd
/var/lib/postgresql
postgres@satnms7-dev:~$ createuser --interactive --pwprompt
Enter name of role to add: satnms
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
```

5.4.6.2 Create SatDB database

New databases can be created via CLI or with SQL statements. This manual uses the CLI tool. There is no difference between database actions or CLI command, createdb is just a wrapper around CREATE DATABASE

- Database must be named `satnms` and the owner of this database must be the role `satnms` as created above
- before you try to to create the database, make sure there exist no database with the same name.
- CLI: https://www.postgresql.org/docs/current/app-createdb.html
- create database: `createdb satnms "sat-nms MNC database"`

example output

```
TODO
```

5.4.6.3 Load SatDB database schema

To initialize the newly create database with all the required object definitions (schema) the installed `migrate` tool will be used. This tool needs the schema definition files `*.sql`) which are installed by the `satnms4-satdb` package to `/home/satnms/satdb/` . Usually this package automatically runs the database migration steps, but if this fails you will find below how to run the step manually.

You can check the current database version with:

```
migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=disable' -path
```

- copy migration files to `/home/satnms/satdb/`
  - for each migration (aka database version) two files exists
    - `[number]_init_satdb.up.sql`
    - `[number]_init_satdb.dow.sql`
- run migration script to apply the changes.
  - `up` command migrates to the latests version
  - execute this as user `satnms`, its not necessary to switch to user `postgres` because the database and role `satnms` are already created in the previous steps.
  - `migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=dis`

Remark: If the database is already initialized it will only run the migration for the new version(s). E.g. if already database version 2 is installed and there are migration files for version 3 and 4 it will upgrade the database to version 4.

example output

```
satnms@satnms7-dev:~$ migrate -database 'postgres://satnms:satnms@localhost:5432/satnr
1/u init_satdb (81.381471ms)
2/u init_satdb (96.040331ms)
satnms@satnms7-dev:~$
```

5.4.6.4 Force database version

In case a migration step fails, migrate will not let you run other migrations on the same database and will mark the database as dirty. You have to fix the migration and then force the corresponding version.

- run `migrate -database 'postgres://satnms:satnms@localhost:5432/satnms?sslmode=disable'`
- replace [number] by the database version you want to force. e.g. `1`

5.4.6.5 Delete database user

If you created a user by mistake you can remove it. For details refer to the PostgreSQL manual:

- CLI: https://www.postgresql.org/docs/current/app-dropuser.html
- `dropuser` is a wrapper around `DROP ROLE`

example output

```
postgres@satnms7-dev:~$ dropuser --interactive satnms
Role "satnms" will be permanently removed.
Are you sure? (y/n) y
```

5.4.6.6 Remove database

If you created a database by mistake you can remove it. For details refer to the PostgreSQL

manual:

- CLI https://www.postgresql.org/docs/current/app-dropdb.html
- `dropdb` is a wrapper around `DROP DATABASE`

6 Theory Of Operation

This section gives some background information about the internal structure and some functional concepts of the M&C software. It may be helpful to know how things work under the hood - both for operating and especially for configuring the software.

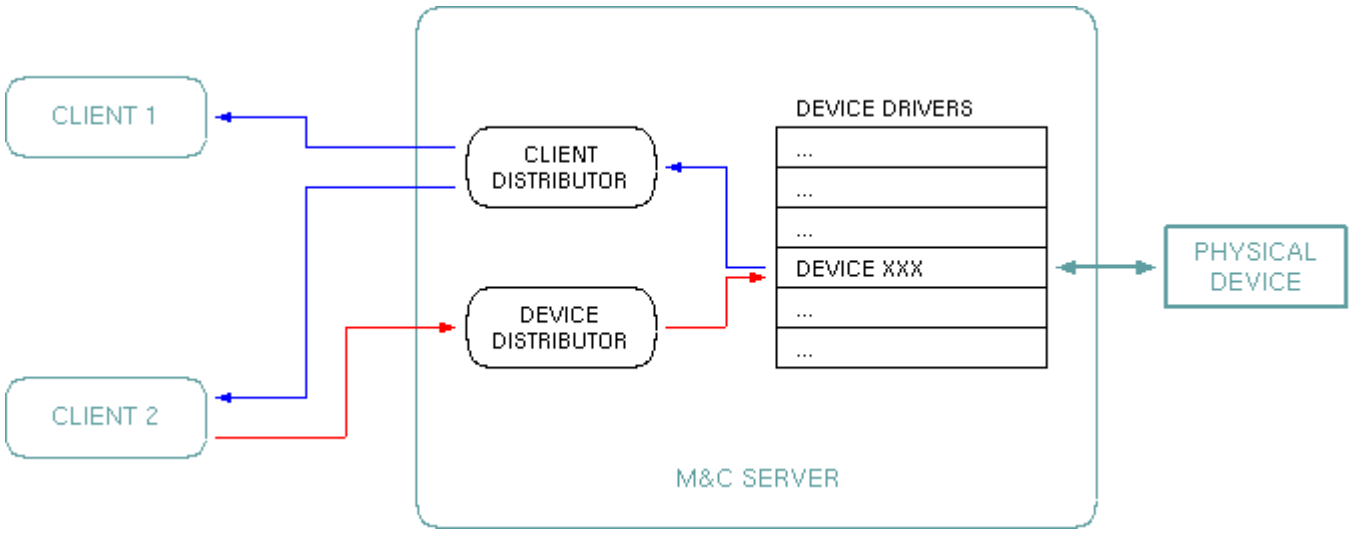
- Chapter M&C Concept describes the basic concepts how a parameter setting at a particular device finds its way to the user interface, how device drivers work and how the M&C software handles the data read from and sent to a device internally.
- Chapter Event Logging describes the functions of the event log database in the M&C software. Event logging happens in a persistent, rotating database of a fixed size (in days of stored events), it supports storing the event logs of multiple M&C systems at a single place.
- Finally, chapter Automating Procedures With Macros gives some information woh the macro feature in the M&C software works as well as examples how to edit macros in the editor or write the mfrom scratch.

All this information refers to the legacy java client of the software but it applies to the web based user interface as well.

6.1 M&C Concept

The following paragraphs give a short introduction to the M&C software's architecture and explains some basic concepts of it. Here a digest in advance:

- The M&C software is built in a client - server architecture. The server dealing with the direct equipment control is a stand alone program running on that machine providing the hardware interfaces which control the equipment. Apart from this, the user interface is another program talking to the server by messages over an TCP/IP channel. A user interface program (client) may run on the M&C server itself or on any other machine having a network connection to the M&C server. There may be multiple clients connected to the same server at a time.
- As mentioned above, messages are used for the communication between the client and the server. Moreover, most of the internal communication between the server's components are done by messages, too. Messages report parameter settings or status information when sent to a client and the transport commands the operator enters at the user interface to the server. Chapter Parameter Messages explains this messaging mechanism more detailed.
- Within wide parts of the software, equipment settings are treated as abstract values. The M&C server uses pluggable modules called 'device drivers' to translate these values to commands the physical device understands. Device drivers are user configurable and user extensible, making the software highly adaptable. Chapter Device Drivers explains this device driver concept



To deepen the explanations given above, this diagram shows a simple configuration with two clients (user interfaces) connected to a M&C server. With this example the flow of information shall be explained.

The first situation which shall be shown, is the polling of status information from the controlled device. The M&C software (more precisely: the device driver) polls all status information and all equipment settings on a regular basis. Control parameters like a frequency or data rate setting are polled less frequently, as they are not expected to change by themselves. At this point we will look to a frequently polled parameter like a lock alarm flag. The device driver polls this flag once a second. With each cycle the following actions are performed:

1. The device driver sends a command string to the device, reads the response and decodes the 'lock alarm: yes/no' information from the device's reply.
2. The device driver codes this information to a parameter message and forwards the message to the 'message distributor'.
3. The message distributor looks in it's message memory for the previous value of this message. If there is a previous message and it contains the same state of this lock alarm, nothing happens. The distributor suppresses multiple messages with the same contents. If however the value has changed, the distributor remembers the new value and sends a copy of this message to every client which is interested in this parameter. In this case both clients will receive a message with the new lock alarm state.

Changing an equipment setting on an operator's request happens in a similar way. After the operator entered the new value at the user interface, the following steps are made by the software:

1. The client software codes the value entered by the operator to a message. It sends this message to the M&C server.
2. At the server, the 'device distributor' receives this message and routes is to the device driver of interest.
3. The device driver receives the message and notes it down.
4. Synchronized with it's polling activities, the driver picks up the changed value and translates it to a command string understood by the physical device. It sends this

command string and processes the device's acknowledgment.

5. Now the device driver triggers an extra polling cycle for the parameter it just has written. This way the parameter is read back and distributed in the same manner described above for the lock alarm fault flag.

6.1.1 Parameter Messages

Parameter messages are one of the basic concepts of the software. They are used to transport parameter settings and status information between the components of the software.

A Message is composed of a message identifier and the value which shall be transported. The latter can be a text string, an enumerated value, a numeric value (integer or floating point) or a boolean flag. The message ID indicates the source or destination of the message.

Message identifiers

The message identifier defines the device driver variable, this message belongs to. Depending on the direction in which the actual message is sent, this can be the recipient of the message or the originating variable.

A message identifier is a textual string composed of two or more words which are separated by dots. The first of these words is the name of the device, the remaining part of the identifier is the name if the device driver variable. A message identifier ' **MODEM-1.tx.on** ' refers to the variable ' **tx.on** ' at the device '**MODEM-1** '.

The usage of messages for parameter settings

Picking up the example with the '**MODEM-1.tx.on** ' parameter, the following shall explain what happens, if a parameter is changed by the operator. The user interface window containing this parameter sends a message to the M&C/VLC server which dispatches the message to the device driver variable which listens to messages with this identifier. The device gets the new value and sets it at the device. The value will be read back (immediately after it has been set, but in regular intervals, too) and a message with the value read from the device is distributed to all windows showing this parameter.

Range messages

Well, but how does the user interface 'know' that '**MODEM-1.tx.on** ' is a CHOICE parameter with the possible selection OFF or ON?. It does not know this from it's own, it receives a so called *range message* from the device driver with this information. A range message describes the data type and the valid range of a parameter. It uses the message ID of the parameter it describes with a ' **.R** ' appended. The range message for the parameter above would use the identifier ' **MODEM-1.tx.on.R** '. The message contents describes the '**MODEM-1.tx.on** ' parameter as a (OFF/ON) choice parameter.

A device driver distributes range messages for all variables it maintains. It tells the user interface about the presence type, and range of all it's parameters. Two important consequences of this fact are:

- The user interface screen dealing with a certain device type, can be used for more than one variants of this type, as the parameter ranges (bit rate ranges, clocking options or something else) are defined on runtime by the device driver rather than by the user interface screen itself. The latter acts as a template which gets filled with the range definitions of an individual device.
- A device driver is not limited to distribute range messages once at start up. The driver may change the range of particular parameters dynamically on the state of other parameters. A common example is to enable/disable certain parameters completely, depending on the presence of some hardware options a device provides.

Message transport

Messages are transported within the software over TCP/IP streams. They can be sent within the same software component, from one sub-component to the other. But they also carry their information between the M&C/VLC server and the user interface program (client) which may be on another computer located somewhere completely different.

To to reduce the number of messages which are sent, the software uses some Optimizations, two of them are listed below:

- Messages are distributed only to the recipients which explicitly requested for them. This is done with a subscription mechanism: If for example a window gets opened which shall display the ' **MODEM-1.tx.on** ' parameter mentioned above, the window *subscribes* for the reception of this parameter. The window in turn gets a message with the actual value of this parameter and -- every time this parameter changes -- another message which updates this parameter. When the window closes, it voids this subscription.
- The message distribution agents in the software suppress messages which are identical. Hence, if a Eb/No value gets polled at a modem once a second, the user interface will get this message only if the value changes.

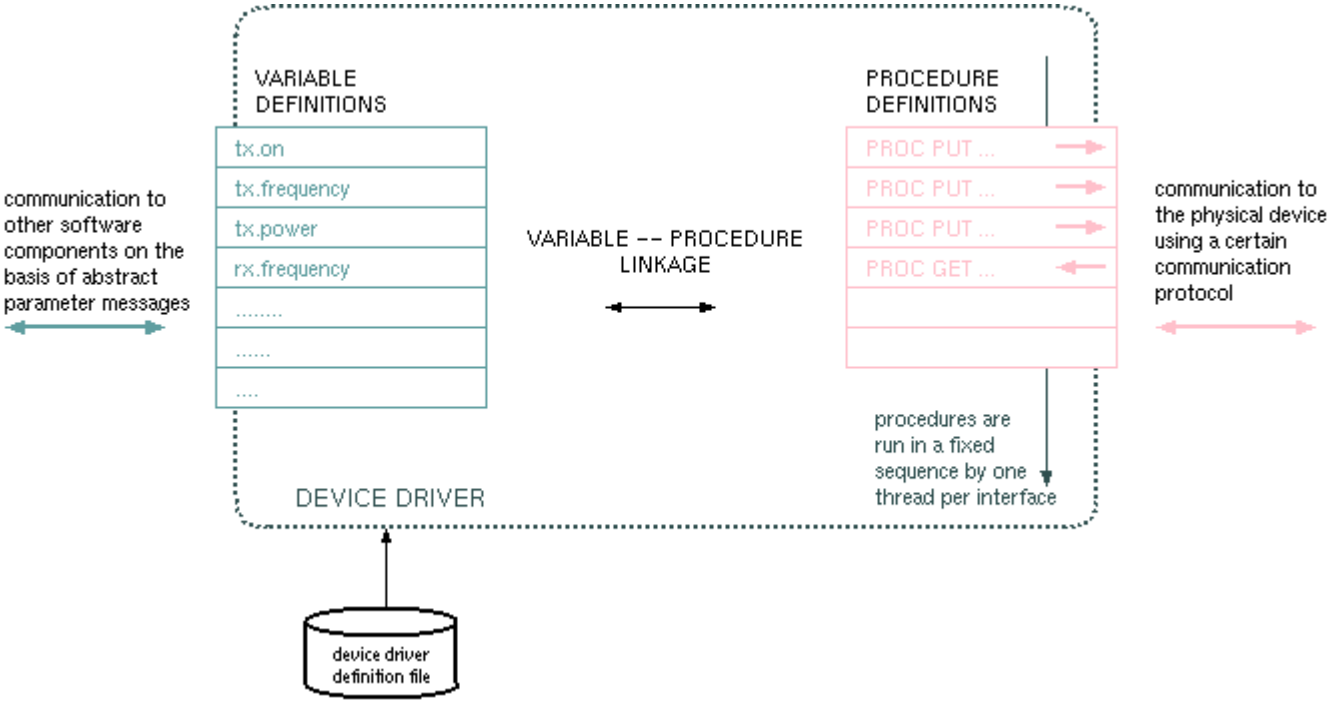
6.1.2 Device Drivers

Device drivers in the M&C software translate abstract parameter settings which are represented by parameter messages into commands sent to the physical device and vice versa. The basic idea is to modularize the software in a way, that one device in a station setup can be replaced by another model, perhaps even by a model made by another vendor, simply by selecting another device driver.

The sat-nms M&C advances this concept by introducing a 'universal device driver' which is completely user configurable. The configurable device driver let's you write your own device drivers for device models which are not yet supported by the software. Most of the device drivers coming with the software are built on top of this configurable driver, so there are a lot of examples you can use as a template.

The universal device driver

The figure below illustrates the structure of a device driver built with the 'universal device driver'. The principal function, however, applies to 'hard coded' drivers, too.



To other software components (primarily to the user interface) the device driver interfaces by a list of variable definitions. On the other hand, a list or procedures does the 'real work', translating the abstract parameter values to physical command sequences. In between a function here called 'variable -- procedure linkage' defines which procedure is run to set or read a certain parameter.

With the universal device driver, all this is setup (compiled) during the program initialization from a text file which describes the device driver in special, but quite simple programming language.

Variables

Variables are the interface between the device driver and the other components of the software, specially the user interface. Each variable acts as an end point for a parameter message. It may receive messages -- which causes the driver to set this parameter at the physical device -- but also may send it's parameter message in order to tell the user interface about the actual setting of this parameter. Chapter ' Device Driver Variables ' explains this more detailed.

Procedures

There are two types of procedures a driver may define. A 'PUT' procedure sends one or more parameters to the physical device after coding them into the format the physical device expects. A 'GET' procedure sends a request to the device, reads the reply and decodes one or more variable values from the data received. A procure never can be both, 'PUT' and 'GET' at the same time.

The driver's environment in the software provides one program thread for each physical interface (e.g. RS232 port). Within this thread all procedures of all device drivers communicating through this interface are called in a fixed sequence. The threads are running concurrently, utilizing the computer's capacity for one thread while another is waiting for a device's reply. Chapter ' Device Driver Procedures ' gives more information about how procedures work.

Special types of device drivers

Not all device drivers are implemented by means of the universal device driver. On the one hand there are a couple of devices which use such a trick method for their remote control, that the universal device driver is not capable in handling this.

The other and larger group of 'hard coded' drivers comprises the so called *logical devices* . These devices do not communicate directly with a physical device, they in fact combine parameter messages in some way and send the result to other device drivers. Power control loops or a software based automatic redundancy switching device are two common examples for this driver family.

6.1.3 Device Driver Variables

Variables are the interface between the device driver and the other components of the software, specially the user interface. Each variable acts as an end point for a parameter message. It may receive messages -- which causes the driver to set this parameter at the physical device -- but also may send it's parameter message in order to tell the user interface about the actual setting of this parameter.

Inside, a variable separately stores two values: The value which recently has been commanded and the value which has been read from the device during the previous polling cycle. By managing these values separately, the device driver is able to check if a value has properly been set by the device. If you see messages like 'Parameter some.name set to X but reads Y' in the event log, then the parameter polling which followed a command returned a value different from the commanded one.

6.1.4 Device Driver Procedures

The software basically knows two types of procedures: A 'PUT' procedure sends one or more parameters to the physical device after coding them into the format the physical device expects. A 'GET' procedure sends a request to the device, reads the reply and decodes one or more variable values from the data received. A procure never can be both, 'PUT' and 'GET' at the same time.

As mentioned in chapter 'Device Drivers' , all procedures of the driver are called on a regular basis. The procedures are called in the same sequence as they appear in the driver definition file. Procedures which actually have no work to do, are skipped by the driver.

A procedure links to one or more variables, the 'WATCH' statement in the driver definition file defines the variables a procedure shall link to. Thereby, a procedure may link to multiple variables but each variable may be 'watched' at maximum by one 'GET' and one 'PUT'

procedure. If now one of the 'watched' variables receives a new value an operator wants to set at the device, the variable marks the 'PUT' procedure which it is linked to as 'dirty'. With the next cycle, the driver recognizes this flag and does *not* skip the procedure this time.

Hence, setting the parameter at the physical device by the procedure happens not synchronously when the parameter message arrives. The physical equipment setting is done decoupled from the arrival of the message, but synchronized with the polling activities the device driver does on a regular basis. This ensures, that the device driver always finishes any pending polling command before it sets a parameter at the device.

While the method of a dirty flag for a 'PUT' procedure is a quite simple rule to determine which procedure shall be run and which shall be skipped, things are a little bit more complicated with 'GET' procedures. There also is a 'WATCH' statement for 'GET' procedures, but the dirty flag is managed in a more sophisticated way here:

1. When the program starts, all 'GET' procedures wake up with the dirty flag set. The driver has to inquire all parameters from the device when it starts.
2. The same applies after a communication failure. If the device didn't answer for some time, an operator might have replaced a faulty device against a new one which will have different settings. The device driver marks all 'GET' procedures as dirty if a communication failure occurs.
3. Each variable contains a timer, programmed for CYCLE interval, the driver description file individually defines for each variable. If the timer expires, the 'GET' procedure watching this variable is marked dirty. This way, each variable is read on a regular basis using an individual polling cycle.
4. Once a variable has been written to a device in a 'PUT' procedure, the 'GET' procedure watching this variable is marked dirty. This enforces a variable to be read back after it has been written to the physical device. This specially is important with variables having a quite long polling cycle programmed. A read back of the parameter eventually would happen minutes later.

6.2 Event Logging

The sat-nms software provides a elaborate event logging facility which lets the operator keep track of device faults and other events. It stores device fault messages and other events like the start of the software, user logins or the NMS communication connections to VLCs.

The ' Concepts ' chapter below explains how the event logging works with M&C systems and with VLC controlled by a NMS as well. Then the chapters ' The Event Report Window ', ' The Active Faults Window ' and ' The Live Event Log Window ' introduce the user interface to the event logging facility.

6.2.1 Concepts

The event log database in the sat-nms software stores device fault messages and other events. Each individual fault flag a device driver read from the device gets logged when it changes it's state. Beside this events like the start of the software, user logins or an NMS's communication connections to the VLCs are logged.

With device faults, not only the appearance of the fault is logged. Also the end of the fault state is logged by the device driver. The latter message also contains a note about the endurance of the fault state.

Event messages

Each message stored in the event log contains a time stamp, the message source, a message text and a priority value. The event priority tells about the importance of a event message.

- 'Informational' messages are of the lowest priority. Events which do not signal a fault like user logins are logged as informational messages. Also messages which indicate the end of a fault state are marked informational.
- 'Fault' messages (medium priority) signal the fault state of a device or software component.
- The highest priority level, 'Alarm' signal a severe fault which needs to be repaired immediately.

The sat-nms software with it's factory settings does not use 'Alarm'. The operator however can configure the priority of each fault flag in the system to his needs. With a NMS system, event priorities have an impact to the event transport to the NMS (see below).

With a M&C system, the event log is stored at the M&C computer. The user interface software provides a ' Event Report Window ' which lets the operator query the event database and aLive Event Log Window which displays the recent events in a scrolling list.

The event database is limited in size. It implements a circular buffer which automatically purges all events older than a configurable limit. The factory setting for event log depth is 365 days. The M&C or NMS system administrator may change this by modifying a configuration file.

Collecting events in the NMS

With a sat-nms network management system, the event log at the NMS site centrally stored the messages originated from all VLCs in the network. Each particular VLC uses a temporary buffer to hold it's events messages until it is able to send them to the NMS. In this case the event log stores additional data with each event:

- The originating VLC is stored as a second source field in each record.
- The time when the message arrives at the NMS is stored as a second time stamp.

The event queue in the VLC

Each VLC contains an event queue which intermediately stores event messages until they are sent to the NMS. The ' System ' device which is present in each VLC controls the behavior of this queue. It is essential to configure the parameters of an VLCs event queue when the VLC is setup, event messages may be lost if they are left unconfigured. This also applies to a M&C system, here the event queue must explicitly must be switched off by setting 'localhost' as destination and all threshold values to zero.

6.3 Automating Procedures With Macros

The sat-nms software includes a powerful macro recording and playback feature which lets you automate complex equipment settings to one mouse click. This feature is built on the parameter message concept of the software. Recording a macro means recording the parameter messages you sent when you change setting at the user interface.

The software stores recorded macros as plain text files, using a simple format which is easy to understand. This and some additional commands in macros, called 'macro directives', make the macro facility to a powerful tool:

- Macros may be edited for reworking or to create simple ones from scratch.
- Macros may be nested. Calling macros from inside macros lets you structure your macros in 'subroutines'
- Macros may contain 'formal parameters' which are substituted while the macro is executed.
- The execution of macros may be bound to buttons in user configured screens.

The following pages describe how macros work and how you use the functions in the software managing macro recording and playback.

6.3.1 Concepts

The M&C/VLC software uses parameter messages to transport equipment settings from the software's operator interface to the physical piece of equipment. Recording a macro simply means recording these messages to a file. Playing a macro sending again the messages from the recorded file.

Editing macros

A recorded message sequence is stored as a plain text file. This makes it easy to rework a recorded macro or even to write a macro from scratch. In a macro file all empty lines or comment lines (starting with a "#" character) are ignored. The other lines either contain equipment settings or macro directives .

Coding of equipment settings

In a macro, equipment settings always appear in the form

<message identifier> <value>

The message identifier consists of the name of the device which shall receive this setting followed by a dot and the parameter name to set. *MOD-2.tx.on* for example addresses the parameter 'tx.on' at the device named 'MOD-2'. A list of the parameter names recognized by the device types can be found at section ' Device Driver Index '. The device name in the macro may be an asterisk ('*') to indicate that this message shall be sent to all devices.

Macros vs. Device Presets

Macros and device presets both help you to automate extensive equipment settings. They work, however, completely differently:

- A device preset records the *state* of a device with all it's settings.
- A macro records the *changes* you apply to the device's settings.

To be sure that a device with all it's parameters is set to a certain state by a macro, you may apply a preset to the device while you record the macro. The macro will contain all settings from the preset.

6.3.2 Macro Directives

Beside plain definitions of parameter settings, a macro definition may contain a couple of additional commands called 'macro directives'. The macro processor recognized these directives and handles them separately. Directives all are lower case words starting with a percent character. The following directives are recognized:

%def

The %def directive defines a substitution parameter. The macro processor knows ten substitution parameters called 0..9. They work similar to parameters in shell scripts (Linux) or batch files (Windows). Any sequence of a percent character followed by a digit (%0 .. %9) in the macro gets substituted by the value defined in a %def directive before. The syntax of the %def directive is

%def [0..9] *substitution text*

For example, ' **%def 4 KSTAR-2** ' sets the substitution parameter '4' to the value 'KSTAR-2'. The character sequence '%4' gets replaced by 'KSTAR-2' in all equipment settings the follow this line in the macro. The substitution parameters are initially empty.

%inc

The %inc directive includes another macro definition at it's place when the macro is played. The syntax of the %inc directive is

%inc *macro-name*

You may nest the execution of macros up to eight levels deep using this directive. If the macro you reference does not exist, the directive is silently ignored.

A common practice of using included macros is to record basic equipment settings into a set of macros. Using the macro editor you can then create "super macros" which first call one ore more of the basic macros and finally set a couple of settings like the channel frequency or the transmit power explicitly.

%wait

The %wait directive lets you pause the execution of a macro for some time. The syntax of the %wait directive is

%wait *msecs*

where *msecs* is the time to pause in milliseconds. The paragraph 'Timing Issues ' below

explains why delays in the macro execution may be helpful.

6.3.3 Timing Issues

When executing a macro which sends equipment settings to more than one device, you might expect that the parameters are set in the same order they appear in the macro definition. In fact this is *not* the case, due to the parallel concept of the M&C software. The following example explains this issue:

```
#
# example macro definition
#

# switch off the carrier to avoid spurious transmissions
#
EIRP.on "OFF"

# turn the modulator to 8 MBit
#

# switch the carrier on again
#
EIRP.on "ON"
```

The example above which tries to clamp down the transmission while the modulator switches to another bit rate, definitely will not work as expected. The macro processor will send the settings in the order they appear to the M&C /VLC server. The latter receives the settings and passes them to the device drivers. Each device driver contains a buffer for the commands pending for execution. The driver for the EIRP setting and the modulator driver work in parallel. The driver for the TX-ON setting takes down the transmit power and immediately brings it up again. It does not wait until the modulator driver has finished the settings.

To overcome this problem, you should slow down the execution of the macro and insert %wait directives at the critical steps. Below the same example with programmed delays:

```
#
# example macro definition
#

# switch off the carrier to avoid spurious transmissions
#
EIRP.on "OFF"

# wait until the transmission is off
#

# turn the modulator to 8 MBit
#

# give the modulator 3 seconds to stabilize with the new bit rate
#

# switch the carrier on again
#
EIRP.on "ON"
```

7 Device Driver Reference

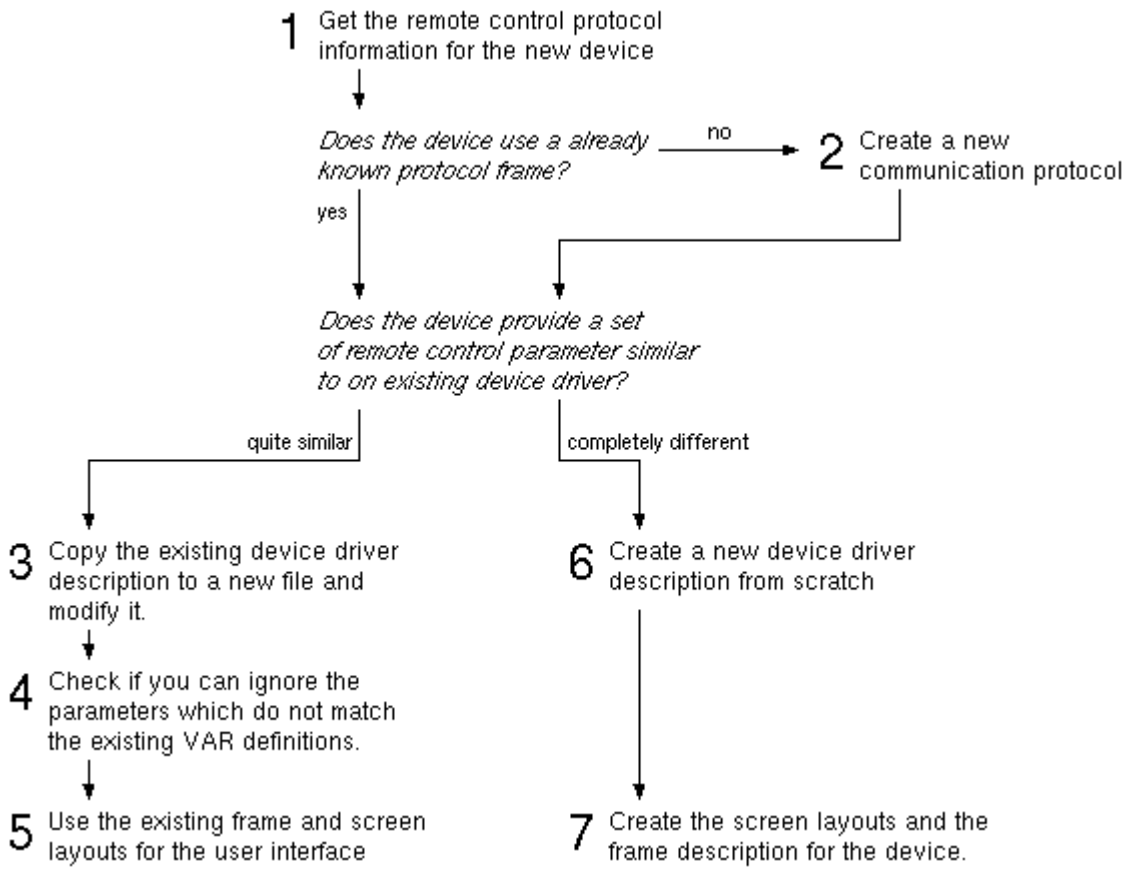
Device drivers in the MNC software translate abstract parameter settings which are represented by parameter messages into commands sent to the physical device and vice versa. The basic idea is to modularize the software in a way, that one device in a station setup can be replaced by another model, perhaps even by a model made by another vendor, simply by selecting another device driver.

The ***sat-nms*** MNC advances this concept by introducing a **universal device driver** which is completely user configurable. The configurable device driver let's you write your own device drivers for device models which are not yet supported by the software. Most of the device drivers coming with the software are built on top of this configurable driver, so there are a lot of examples you can use as a template.

To make the ***sat-nms*** software support an additional device type, the software needs at least three components:

1. A low level communication protocol which handles the protocol frame including device addressing or checksum calculation.
2. A device driver which defines the parameters the operator may inspect or control at the device. The device driver also contains the I/O routines to exchange these parameters with the device.
3. Finally, there must be a standard user interface (the so calleddevice oriented user interface) for the new device which lets an operator view or modify the device parameters.

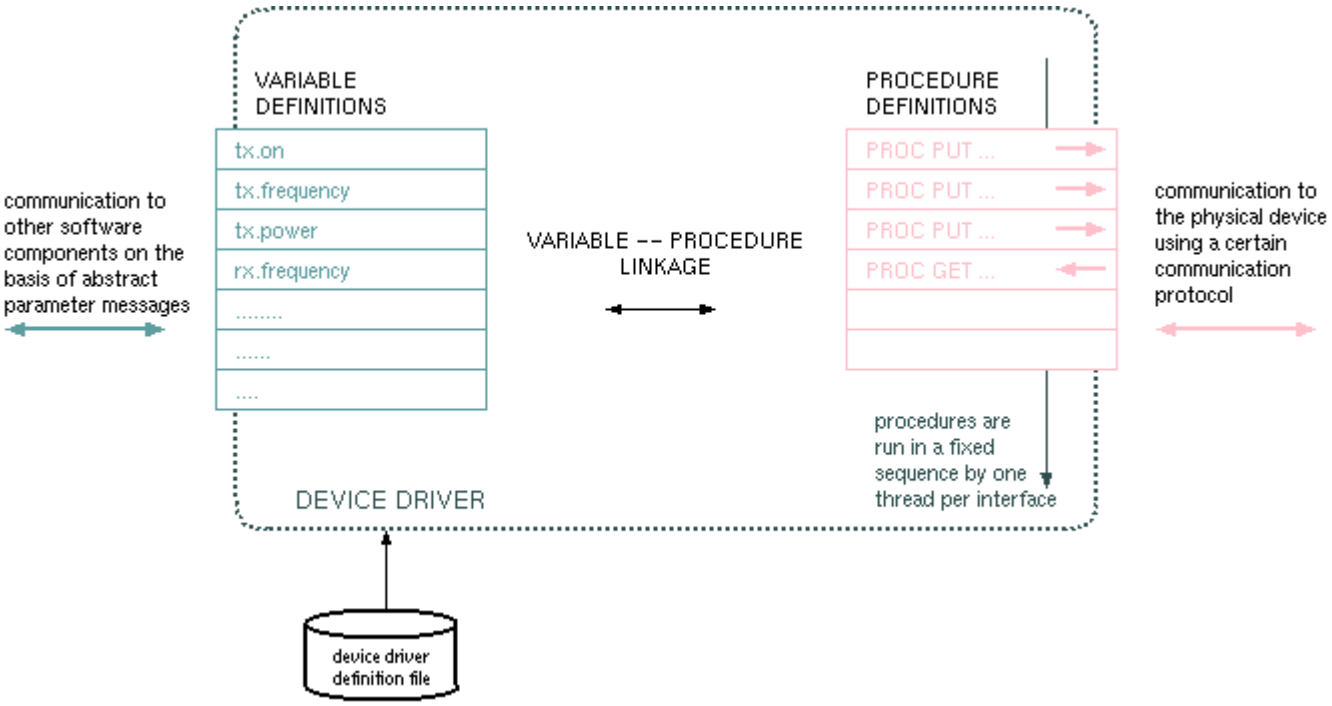
With the ***sat-nms*** software, all three components from the physical M&C interface at the device up to the representation of parameters on the screen are configurable and extensible by the customer. The ***sat-nms*** software comes with library supporting a large number of devices. These files are at your's disposal to use them as templates or as ready made module for the device driver you need to develop.



The diagram above illustrates the first steps to the integration of a new device type into the ***sat-nms*** software. You are encouraged to use as much as possible of the existing protocol, driver and user interface definitions.

7.1 Understanding the device driver

The figure below illustrates the structure of a device driver built with the**universal device driver**. The principal function, however, applies to 'hard coded' drivers, too.



To other software components (primarily to the user interface) the device driver interfaces by a list of variable definitions. On the other hand, a list of procedures does the 'real work', translating the abstract parameter values to physical command sequences. In between a function here called *variable -- procedure linkage* defines which procedure is run to set or read a certain parameter.

With the universal device driver, all this is setup (compiled) during the program initialization from a text file which describes the device driver in special, but quite simple programming language. The following pages describe how variables , procedures and the protocol encapsulation work together.

7.1.1 Variables

Variables are the interface between the device driver and the other components of the software, specially the user interface. Each variable acts as an end point for a parameter message. It may receive messages -- which causes the driver to set this parameter at the physical device -- but also may send it's parameter message in order to tell the user interface about the actual setting of this parameter.

Inside, a variable separately stores two values:

- The value which recently has been commanded and
- the value which has been read from the device during the previous polling cycle.

By managing these values separately, the device driver is able to check if a value has properly been set by the device. If you see messages like `Parameter some.name set to X but reads Y` in the event log, then the parameter polling which followed a command returned a value different from the commanded one.

7.1.2 Procedures

Driver procedures actually perform the communication with the device to control. They operate the device as commanded and poll the equipment settings and state.

There are two types of procedures a driver may define.

- A `PUT` procedure sends one or more parameters to the physical device after coding them into the format the physical device expects.
- A `GET` procedure sends a request to the device, reads the reply and decodes one or more variable values from the data received.

A procedure never can be both, `PUT` and `GET` at the same time.

The device driver executes the defined procedures in an endless loop, called the polling cycle. But a procedure is not necessarily called in every cycle. If there is nothing to do for a particular procedure, it is skipped. To determine which procedure must be executed in a cycle, procedures are bound to variables. A procedure may be bound to one or more variables, however, there may me at most one `PUT` and one `GET` procedure referencing a variable.

A `PUT` procedure is executed, if at least one of the variables it is bound to has received a new setting, e.g. from the user interface. For a `GET` procedure the following conditions cause the procedure to be executed:

1. The device driver has established communication to the device after power-on or after a communication interruption.
2. The (individually defined) polling interval for at least one of the variables bound to this procedure has elapsed.
3. At least one of the variables bound to the procedure has been commanded to the device. The parameter must be read back for verification.

7.1.3 Protocol encapsulation

The ***sat-nms*** device driver concept encapsulates the protocol frame used for the communication with a device in a separate layer. This protocol layer describes which kind protocol frame has to be wrapped around the commands sent to a device. There are a number of important advantages with this concept:

- The protocol frame is automatically added to each command or request sent to the device. The other way round, the driver automatically strips of this 'envelope' from the data received from the device.
- A certain type of protocol frame needs to be programmed only once and may be used for several device drivers, if for example different devices from the same vendor use one and the same protocol definition.
- Some devices may be operated either by one or the other communication protocol. The protocol encapsulation allows easily to switch between the variants.

With the ***sat-nms*** software, new protocol definitions may be added to the software, simply by editing a text file for the new type of protocol frame which is needed.

If you are going to write a new device driver, you first should investigate if there is an existing protocol definition in the ***sat-nms*** library which can be used to serve the new device. If no protocol definition matches, the chapter Device communication protocolsdescribes to write a new protocol definition.

7.2 Writing a device driver

As mentioned above, device drivers are coded as simple text files. When the MNC program starts, it 'compiles' the device drivers needed for it's equipment setup to memory. Writing a device driver means editing such a text file and storing it at a place in the MNC computer where the MNC program searches for device drivers.

The ***sat-nms*** device driver language has been designed to be very easy to understand. If you have a look at the device driver files coming with the ***sat-nms*** software, you probably will understand most of this language after an hour or two.

A MNC or VLC system keeps all device drivers in a subdirectory called `drivers`. On standard installations this is the directory `/home/satnms/drivers`. The name of the device driver files consist of the driver name (usually something like `Manufacturer-ModelNumber`) followed by the extension `.device`.

If you are writing the device driver on a Linux based MNC or NMS computer, you may want to use the VI text editor or mcedit (included in midnight commander) text editor for this. Both has been configured to colorize device driver files on these machines. We also providing syntax definitions for Visual Studio Code. Please ask your contact at SatService for details.

You also may copy device driver files to a MS-Windows based computer to edit them there. If you do this, you should consider the following:

- Device driver files are Unix based text files. Lines are terminated with a line-feed character only. Your favorite MS-Windows text editor may have problems to show these files.
- Unix / Linux is case sensitive with file names. Be sure that you don't mess up the case of characters in file name when you copy files between Unix and MS-Windows.

7.2.1 Starting from scratch

The following pages describe the 'hard way' to create a device driver from scratch. It is a good exercise to do this once for a simple driver. In practice however, in most cases you will use an existing driver as a template and modify this for your needs.

The basic steps to build a device driver from scratch are:

1. Naming the device driver
2. Selecting the communication protocol
3. Including the standard definitions file
4. Defining the driver variables
5. Adding the data exchange procedures

If you are reading this manual online, you may open theNDSatCom-KuBand-Upconverter driver example in a separate window to watch this in parallel while you are reading the following pages.

7.2.1.1 General file format

The ***sat-nms*** device driver language has been designed to be very easy to understand. The syntax resembles the BASIC programming language in some aspects, however, the language is highly specialized for it's purpose. Here some basic definitions which help you to read or write device driver files:

- The device driver language is case sensitive for all identifiers and keywords.
- Whitespace (space characters, tabs, line breaks) separates words.
- Line breaks have no special termination function.
- All keywords (except the RPN commands) are in upper case letters.
- Comments in C/C++ style are recognized (both, `/* ... */` and `// ...` comments).
- Identifiers (names for variables and tables) may consist of letters, digits and dots. They must start with a letter.

Beside the rules listed above, the device drivers provided by SatService GmbH follow some sort of coding/formatting convention:

- Each file starts with a comment block containing a short description and a change history.
- Identifiers always start with a lowercase letter. If a name consists of more than one word, the first letter of each following word is in upper case. Dots are used to group parameter functionally. Example: `mod.symbolRate` .
- Statements are indented four columns for each level.

7.2.1.2 Naming a device driver

To create a new device driver, the first step is to give the driver a name. The device drivers supplied by SatService GmbH all follow a 2 part naming scheme: `VendorName-ModelDescription` . A SatService ACU2 antenna control unit appears as `SatService-ACU2` in the device driver list. This makes it easy for the person who configures the equipment setup of a MNC system to select the appropriate device drivers.

The name of the device driver appears at several places:

- It is used as part of the device driver file name.
- Exactly the same name should be defined as a initialized variable called `info.type` . The online help function and the device preset directory selection of the user interface depend on this definition. Chapter Info variables gives more information about this.
- The COMMENT statement also contains the device driver name.

The COMMENT statement

The text following the COMMENT statement is free field and principally may contain any information. The device drivers coming with the ***sat-nms*** software all follow a convention which defines the comment string as

Driver-name X.YY YYMMDD

where X is the major version number of the driver, YY the minor version number and YYMMDD the release date of this driver version. It is recommended that customer defined device driver follow this scheme, too.

The comment statement in fact creates a hidden info-variable definition. This variable with the name `info.driver` is initialized with the comment text. At the Info-page of the device oriented user interface you can view the comment text or the driver name /version / date respectively.

Example

```
COMMENT   "NDSatCom-KuBand-Upconverter 1.03 010809"
```

This example, taken from NDSatCom-KuBand-Upconverter device driver, identifies this driver as version 1.03, released at August, 9th, 2001.

7.2.1.3 Selecting a communication protocol

With the ***sat-nms*** software, the device communication protocol handles the low level communication between MNC and device. There are trivial protocols which simply put a newline character at the end of each message and more sophisticated ones dealing with start/end characters, checksums, device addresses and more.

The ***sat-nms*** software comes with a bunch of predefined protocol definitions. If none of these protocol definitions matches, the chapter Device communication protocols describes to write your own protocol definition.

The communication protocol used for a certain device (for a certain serial interface, more exactly spoken) is defined with the equipment setup which you can configure a the graphical user interface. The device driver defines a preferred communication protocol. This means, in the device driver you state the communication protocol this driver is designed for. At the equipment setup screen of the software you still are able to choose a different protocol for a device of this type, however, you will be warned by the software. The preferred communication protocol for a device driver is defined with the PROTOCOL statement:

The protocol definition this statement refers to must exist in the `protocols` subdirectory of the software. On standard installations this is the directory `/home/satnms/drivers`.

Example

```
PROTOCOL  Miteq-MOD95
```

This example, taken from NDSatCom-KuBand-Upconverter device driver, defined the Miteq-MOD95 protocol as the preferred one for the driver.

Some communication protocols may be configured in some aspects of their functionality by applying protocol parameters to them. This is done with the `PROTOCOLPARAMETER` statement:

There may multiple `PROTOCOLPARAMETER` statements in on device driver file, each defining one aspect of the communication protocol. The `key` defines the parameter to set, `value` is the value to be assigned to this parameter. Please refer to the documentation of the protocol you are using (usually given in the comment section of the `*.proto` file) if there are configurable parameters for this protocol and if you need to set them for your application.

Protocol parameters may be set either in the protocol definition file (thus defining the parameter for all uses of the protocol) or in the device driver. A definition in the device driver overwrites the setting in the protocol file.

ATTENTION: If you have multiple devices in the same device thread (`INTERFACE`) using different device drivers, the result is unpredictable if the drivers do not make exactly the same protocol parameter settings.

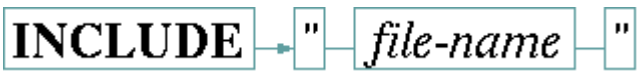
Example

```
PROTOCOLPARAMETER "post.content.type=application/xml"
```

7.2.1.4 File includes

ATTENTION: File including does not work when a device driver file gets interpreted by the *client* software, e.g. when a device preset shall be formatted along the device driver's parameter definitions. Variables defined in included files get not properly formatted in the device preset window. **For this reason it is recommended to use `INCLUDE` only for the standard includes as described in the example below!**

Device driver definition files may include other files. Frequently needed definitions may be encapsulated in included files which makes it easy to change these definitions at a central point. The syntax of the INCLUDE statement is:



The file name given in the INCLUDE statement is relative to the MNC software's home directory. With the ***sat-nms*** Software, there are two standard include files. Almost any driver provided by SatService GmbH includes one of these files. The files are named `Standard.inc` and

`StandardBin.inc` respectively. They define the so called low level interface to the device and the fault flags which indicate a communication failure for the device.

Example

```
INCLUDE   "drivers/Standard.dotinc"
```

This includes the standard ***sat-nms*** include file for driver which use a text based communication protocol. The file is located in the `drivers` subdirectory which is one level below the MNC software's home directory.

7.2.1.5 Tweaking the processing sequence

By default, the device driver executes the `PUT` and `GET` procedures in the same sequence, the procedures appear in the the driver file. Procedures which are not ready to run are skipped, but the overall sequence of execution is given by the order in which the procedure appear in the driver.

For PUT procedures this has the implication, that they are not executed immediately after an operator (or commanding device) has sent a new value for a parameter. The procedure has to wait it's turn. If there are a lot of `GET` procedures ready to run, this may take some time, specially with complex devices which are quite slow in response.

By specifying the `PUTPRIORITY` mode in the header of the device file, you can speed this up, making the device driver for a slow device more responsive:

You enable the `PUTPRIORITY` mode for a driver by adding `PUTPRIORITY` as a top level keyword to the driver file. This changes the order of execution of procedures in the following way:

If at least one `PUT` procedure of the driver is ready to run, all `GET` procedures are skipped. `GET` procedures only are executed if no `PUT` procedure is ready to run. The outcome of this is, that the `PUT` procedure is executed not later than the device thread's `IDLE` time after the new value has been commanded.

But as with most good things there is one hitch with this: If you change a parameter once every driver cycle of faster, `GET` procedures will never get a chance to run. The driver will respond quickly and execute any settings you command, but state information like alarms or meter readings will never be updated. This is the reason why `PUTPRIORITY` is not the default behavior of a ***sat-nms*** device driver.

7.2.2 Defining variables

Variables are the interface between the device driver and the other components of the software, specially the user interface. Each variable acts as an end point for a parameter message. It may receive messages - which causes the driver to set this parameter at the

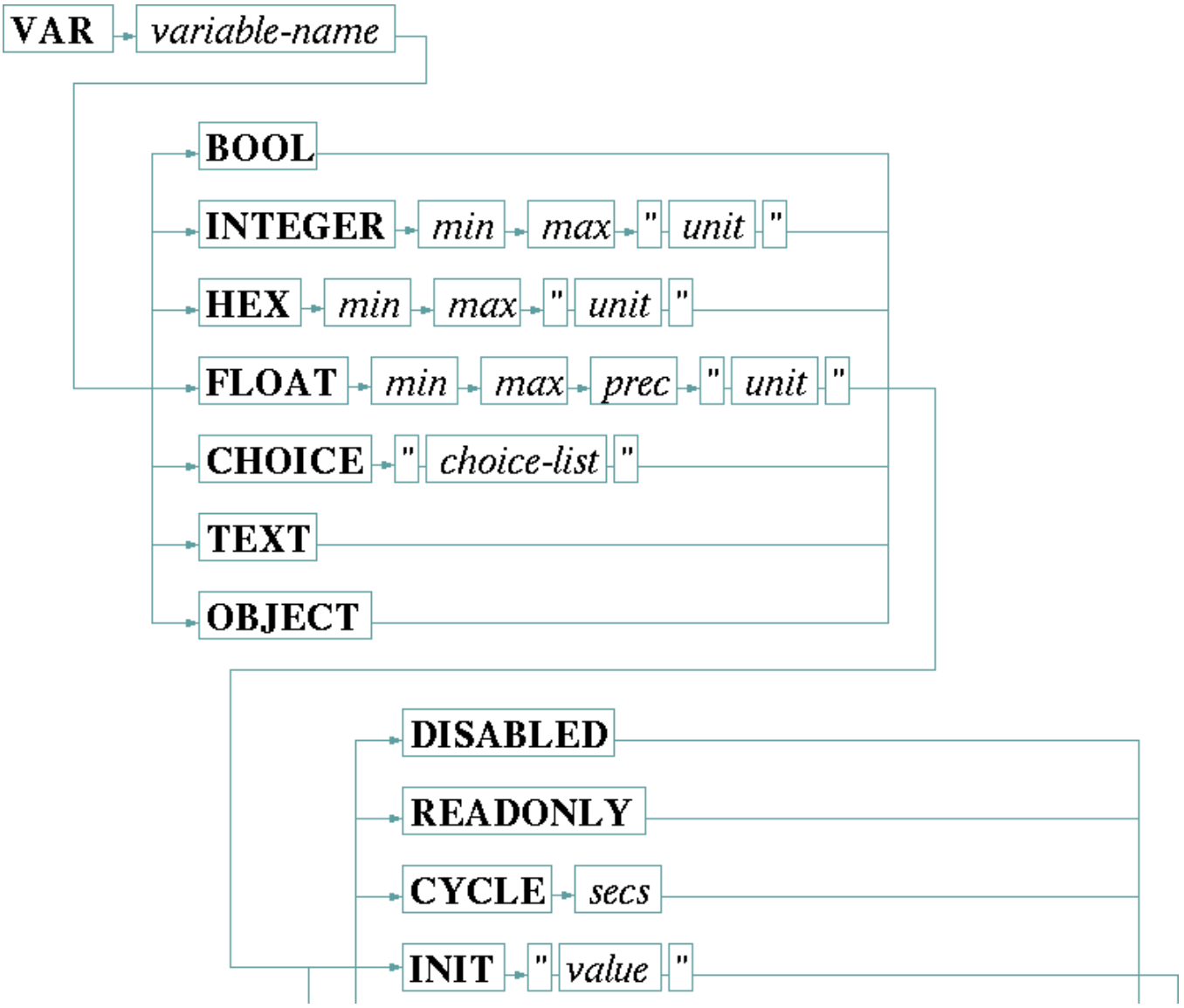
physical device - but also may send it's parameter message in order to tell the user interface about the actual setting of this parameter.

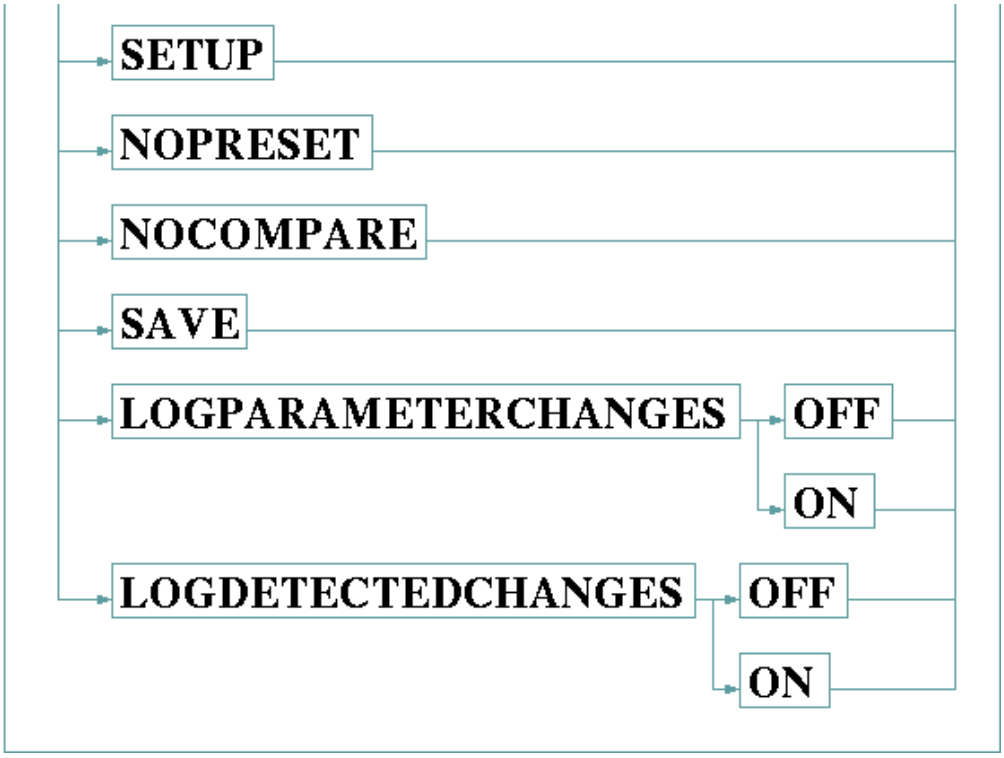
Before you start writing the device driver, you should plan a list of variables which build the software interface to the device. Designing this interface should be done careful, you can save a lot of effort if you are able to re-use variable definitions from other devices. This not only saves the time to write the definitions, it also enables you to use the parameter screens available for this device in the ***sat-nms*** library.

If you intend to write are driver for a device which implements similar functions as an existing driver, it is recommended to copy the variable list from this device.

7.2.2.1 The VAR statement

Device driver variables are defined using the `VAR` statement. With the `VAR` statement you define the name, the data type, the valid range and some other properties of the variable. This is the general syntax for a VAR statement:





A variable definition consists of the VAR keyword, the variable's name, a data type / range definition and optional modifiers which control the behavior of this variable.

Data type / range definition

The ***sat-nms*** device driver knows about 7 data types. The data type of a variable is defined by one of the data type keywords `BOOL`, `INTEGER`, `HEX`, `FLOAT`, `CHOICE`, `TEXT` or `OBJECT`, followed by the range definition for the selected data type. Here the data types listed in tabular form:

BOOL	The BOOL data type can have the values `true` and `false`. the `BOOL` type mainly is used for flags which shall be displayed as signal lamps at the user interface.
INTEGER	The `INTEGER` data type carries numeric integer values (64 bit length). When you define an `INTEGER` variable, you must provide the valid range (`min` / `max`) of the variable and a unit string which is shown at the user interface right of the data. If both, `min` and `max` are zero the user interface software does no range check at all. The unit string may be empty, but the double quotes are required ("").
HEX	The `HEX` data type also is a 64 bit integer, but formatted in hexadecimal notation.

FLOAT	The `FLOAT` data type carries double precision (64 bit length) floating point values. When you define a `FLOAT` variable, you must provide the valid range (`min` / `max`) of the variable, the number of fraction digits and a unit string which is shown at the user interface right of the data. If both, `min` and `max` are zero the user interface software does no range check at all. The unit string may be empty, but the double quotes are required (""). `FLOAT` variables are displayed with a fixed precision. Alternatively you may force a scientific notation by adding 100 to the precision value. Example: `VAR myFloat FLOAT 0 0 103 ""` defines a scientific formatted floating point variable shown with 3 digits precision (e.g. `0.123E-2`)
CHOICE	The `CHOICE` data type defines a parameter which may contain one of a fixed set of string values. You define this set as a comma separated list, enclosed in double quotes. At the user interface such a parameter appears as a drop down box where you can select a value from this list.
TEXT	The `TEXT` data type carries an arbitrary character string.
OBJECT	The `OBJECT` data type is used with complex variables which cannot be shown by the standard user interface routines. `OBJECT` variables only appear together with logical devices, you never will need this data type when writing a device driver.

Modifiers

Modifiers control some properties of a variable concerning it's state, usage, initialization and polling cycle. Modifiers may before or after the type definition.

DISABLED	Variables may be *enabled* or *disabled*. Disabled variables appear grey at the user interface, no value is displayed and no value may be entered. The `RANGESET` command is used to change the enable state. By default variables start in enabled state unless they are marked as `DISABLED` in the VAR statement. Disabling a variable at this point may be useful if the variable stands for a parameter which is not available at all models of the device type the driver is written for. The variable can be enables by the driver when it detects that the device actually connected to the MNC supports this parameter.

READONLY	Marking a variable `READONLY` prohibits the operator from changing it's value. This is used for state variables like meter readings. The range information for the data type used must be provided even if the variable is marked `READONLY`.
CYCLE	The `CYCLE` modifier controls the frequency (expressed as time interval in seconds), this variable shall be polled from the device. By default, variables are read from the device with every working cycle of the driver. This is about one a second if only a few parameters are to read. With many parameters the polling rate will be lower as the response time for each interrogation extends the cycle time.Hence, most device drivers poll only a few parameters like alarm flags or some meter readings with the maximum possible rate. Other parameters, e.g. settings you do not expect to change by themselves, are polled at a much lower rate.Setting the `CYCLE` time to zero causes the driver to do no regular polling for this variable at all. However, after power up and after communication failures, such a variable still is read once from the device
INIT	Using the `INIT` modifier, a variable may be initialized to a certain value. This value remains valid until the variable gets polled the first time. The `INIT` option often is used with variables which are used to configure the driver and never are read from the device itself.
SETUP	The `SETUP` modifier marks a variable to be listed in the maintenance/setup window of the standard device screens. Chapter Setup variables tells more about this special variable type.

NOPRESET	Using the `NOPRESET` modifier, a variable can be explicitly excluded from the set of parameters which are written to a device preset. A device preset contains all variables which define a `PUT` procedure and are no `SETUP` parameters and don't have the `NOPRESET` modifier set. The `reset` variable - if defined - is also implicitly excluded from device presets preset, for backward compatibility reasons. Please note, if you add `NOPRESET` to a variable in an existing device driver, this will not remove the value for this variable from existing device presets, it only will prevent the software from adding the value to new presets. You may use the preset editor to remove the unwanted value manually from existing device presets.
NOCOMPARE	Without the `NOCOMPARE` modifier the device driver will compare a value it commanded to the device against the value it read back after this. If the two values differ, an informational message showing the commanded and the read value is added to the log. `NOCOMPARE` suppresses this check. This is useful with parameters which are known for reading back in a wrong way, frequent messages in the log can be avoided this way.
SAVE	The `SAVE` modifier tells the driver to save this variable on disk and to restore it's value when the program starts. Usually all device settings are stored in the device itself, the MNC system does not change anything at a device when it starts.In some cases, e.g. with `SETUP` variables or if the driver implements a receive level threshold, it is useful to store variable values in the MNC system rather in the device.
LOGPARAMETERCHANGESON/OFF	Defines if this variables shall log changes which are commanded to it. Parameter changes are logged if this is set ON and the `logParameterChanges` switch in the device setup page is activated as well. `LOGPARAMETERCHANGES` is `ON` by default for all variables, hence you only need to state `LOGPARAMETERCHANGES OFF` for variables you want explicitly to be excluded from logging.

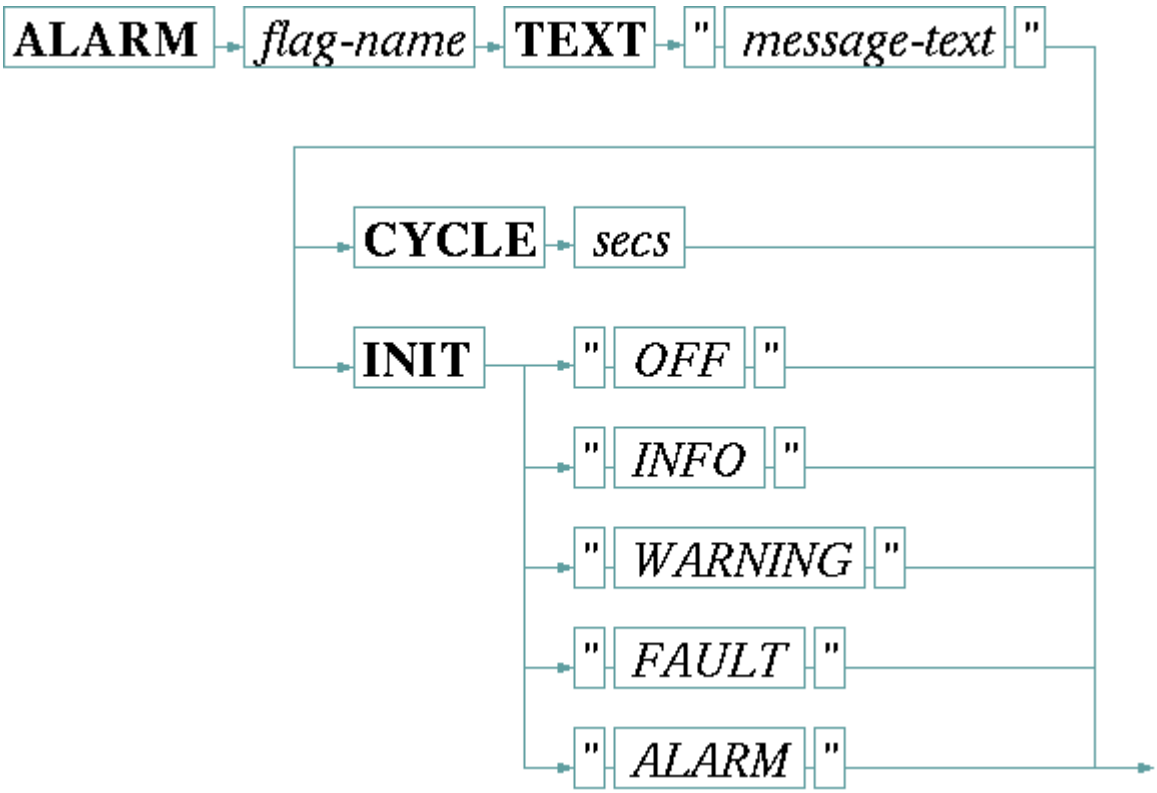
LOGDETECTEDCHANGESON/OFF	Defines if this variables shall log changes which are detected when reading back the values. Detected changes are logged if this is set ON and the `logDetectedChanges` switch in the device setup page is activated as well. `LOGDETECTEDCHANGES` is `ON` by default for all variables, hence you only need to state `LOGDETECTEDCHANGES OFF` for variables you want explicitly to be excluded from logging. Please note, the this modifier has no effect on read-only variables, they are not logged regardless of the logDetected setting.

Example

```
VAR audio.1.dotprogram      CYCLE 0   INTEGER 0 0 ""
VAR audio.1.dotrouting      CYCLE 0   CHOICE "NRM,MON,LFT,RGT"
VAR audio.1.dotoutput       CYCLE 0   CHOICE "ANALOG,AES/EBU,SPDIF,AC3"
VAR audio.1.dotlevel        CYCLE 0   INTEGER 6 18 "dB"
VAR audio.1.dotlanguage     CYCLE 0   TEXT
VAR audio.1.dottest         CYCLE 0   CHOICE "NRM,TEST-1,TEST-2,TEST-3,TEST-4,TEST-5"
VAR audio.1.dotinfo         CYCLE 4   TEXT READONLY
```

7.2.2.2 The ALARM statement

Alarm flags are a special variant of variables. They always are of the type `BOOL` and they are `READONLY`. For each `ALARM` variable you define a text message which appears in the *Faults* page of the device's user interface screen and as a message in the event log if the value of the flag changes. The device driver summarizes all alarm flags with each cycle and generates a summary fault information for the device. Alarm flags are defined with the ALARM statement:



An alarm flag definition consists of the `ALARM` keyword, the flag's name and an optional `CYCLE` definition which work analogous to the `CYCLE` modifier in a `VAR` statement.

The name of an alarm flag must be of the form `faults.XX` where `XX` is a two digit number in the range from 01 .. 98. The `faults.99` is reserved for a communication fault.

The alarm flag definition may include an INIT clause which initializes the fault priority value of the flag. INIT must be followed by one of `OFF`, `INFO`, `WARNING`, `FAULT` or `ALARM`.

Example

```
ALARM faults.01     TEXT "Remote access" INIT "WARNING"
ALARM faults.02     TEXT "Synthesizer"
ALARM faults.03     TEXT "LO-A lock"
ALARM faults.04     TEXT "LO-B lock"
ALARM faults.05     TEXT "Power supply"
ALARM faults.06     TEXT "IF-LO level"
ALARM faults.07     TEXT "RF-LO level"
```

7.2.2.3 Info variables

Each device driver defines a number of so called info variables which tell the operator, but also the client software about the device driver. All info variables

- are named starting with `info.`
- are defined `READONLY` The user interface contains a special screen which display the

contents of all info variables the driver defines.

Info variables are used to display information like device types, serial numbers etc. Beside this, there are three info variables each device driver **must** define in any case. They are:

info.type	This variable tells the user interface name of the device driver. It is used to select the subdirectory for device presets and to find the help screen for this device driver. `info.type` must be initialized to the name of the device driver which in fact is the file name with the trailing `.device` cut off.
info.port	This variable is set by the device driver to the name of the serial port which is used to access the device. This is for convenience, offering the operator this information without opening the device setup.
info.frame	The standard user interface uses the value assigned to this variable to select the 'device oriented' screen set for this device. `info.frame` must be initialized to name of the device frame definition file to be used with this device.

Example

```
VAR info.type          CYCLE 0 TEXT READONLY INIT "Tandberg-Alteia"
VAR info.port          CYCLE 0 TEXT READONLY
VAR info.frame          CYCLE 0 TEXT READONLY INIT "IRD-Alteia"
VAR info.model          CYCLE 0 TEXT READONLY
VAR info.serial        CYCLE 0 TEXT READONLY
```

The example above - taken from the Tandberg-Alteia device driver - identifies the driver as `Tandberg-Alteia` and tells the user interface to use the device oriented frame set called `IRD-Alteia` for this device. The info variables `info.model` and `info.serial` are set by the driver later, when it read the model description and serial number from the device.

7.2.2.4 Setup variables

Another group of variables which are treated specially by the driver are setup variables. They are used to configure the device driver or to set parameters which shall not appear at the standard user interface.

Setup parameters are shown in an own screen which is accessible only to operators of a privilege level of 150 and above. Setup variables must be marked with the `SETUP` modifier, the name of setup variables must start with `.config`.

Example

```
VAR config.lbandInputs    CYCLE 0 CHOICE "4,2" SETUP SAVE
VAR config.lnbPower       CYCLE 0 CHOICE "OFF,ON.,BST" SETUP
VAR config.loFreq         CYCLE 0 FLOAT 0 0 1 "MHz" SETUP
VAR config.berThreshold   CYCLE 0 TEXT SETUP
VAR config.sigThreshold   CYCLE 0 INTEGER 0 255 ""  SETUP
VAR config.errFrame       CYCLE 0 CHOICE "FREEZE,BLACK" SETUP
```

The example above - taken from the `Tandberg-Alteia` device driver - defines a setup variable `config.lbandInputs` which configures the driver for 2- or 4-input models of the IRD. This variable also is marked with the `SAVE` modifier to make the setting permanent. The following variables in the example are configuration settings which are placed in the setup area to keep them out from the common user interface.

7.2.2.5 Variables for internal use

Variables may be used to store intermediate values in the driver. For instance, a bit field status value may be read into a variable and then interpreted using the `BITSET` command.

Variables for internal use must be declared `READONLY` to work properly. Even though not required, it is recommended to give these variables a descriptive name which tells about the internal usage of this variable. The following example is taken from the SSE KStar device driver:

```
VAR internal.state HEX 0 0 " " READONLY   CYCLE 2

...

PROC GET WATCH internal.state
   PRINT "AL"
   INPUT "=" internal.state
   BITSET faults.01 = internal.state 9
   BITSET faults.02 = internal.state 15
   BITSET faults.03 = internal.state 11

...
```

7.2.3 Using conversion tables

`CHOICE` parameter often are implemented by the device's remote control protocol as numeric constants (0 for BPSK, 1 for QPSK) or they are abbreviated in a way which is hard to remember. The user interface of the software shall in contrast display self-explaining names for such settings.

The driver language lets you define translation tables which do such conversions while the driver performs a `PRINT`, `INPUT`, `READ` or `WRITE` statement. As the driver knows, in which direction a table translation must be done, you can use the same table to translate values when they are written to and when they are read from the device.

The TABLE statement described in following chapter is used to define translation tables.

7.2.3.1 The TABLE statement

The driver language lets you define translation tables which do such conversions while the driver performs a `PRINT`, `INPUT`, `READ` or `WRITE` statement. Tables are defined with the `TABLE` statement:

The `TABLE` keyword is followed by a name for this table and the table definition in double quotes. The table definition contains assignments `A=B`, separated by commas. Strings containing commas or equal characters cannot be translated through a table.

```
TABLE tModulation   "BPSK=BPS,QPSK=QPS,8PSK=8PS"
```

In the table definition, the `A` value is the string visible at the user interface of the software, the `B` value is used when communication with the device. The driver translates in a `PRINT` or `WRITE` statement BPSK -> BPS, QPSK -> QPS or 8PSK -> 8PS if the table `tModulation` shown in the example above is referenced. With an `INPUT` or `READ` statement, the table automatically translates the values the other way round.

For some applications a table definition may become quite large. The driver syntax permits to split up the table definition in multiple string/lines. Each string/line must be enclosed in double quotes and must contain at least one translation pair. Commas at the end of a line are omitted. Below an example for such a multi line table definition is shown.

```
TABLE tVideoTest   "NRM=00,625.1=01,625.2=02,625.3=03,625.4=04,625.5=05"
                   "625.6=06,625.7=07,625.8=08,625.9=09,625.10=10,625.11=11"
                   "625.12=12,625.13=13,625.14=14,625.15=15,625.16=16"
                   "525.1=17,525.2=18,525.3=19,525.4=20,525.5=21,525.6=22"
                   "525.7=23,525.8=24,525.9=25,525.10=26,525.11=27,525.12=28"
                   "525.13=29,525.14=30,525.15=31,525.16=32"
```

Remarks

- If a table translation is invoked with a value which is not contained in the table, the translation returns the first value in the table.
- To define a table it may be useful to copy the definition of the `CHOICE` variable which shall be translated at the place of the table statement. this copy then can be extended to the table definition.
- Don't add whitespace characters in the table definition to make this better readable. The driver would treat these characters as part of the strings to translate!

7.2.4 Adding data exchange procedures

While the device driver variables discussed in the chapters above specify the driver's interface

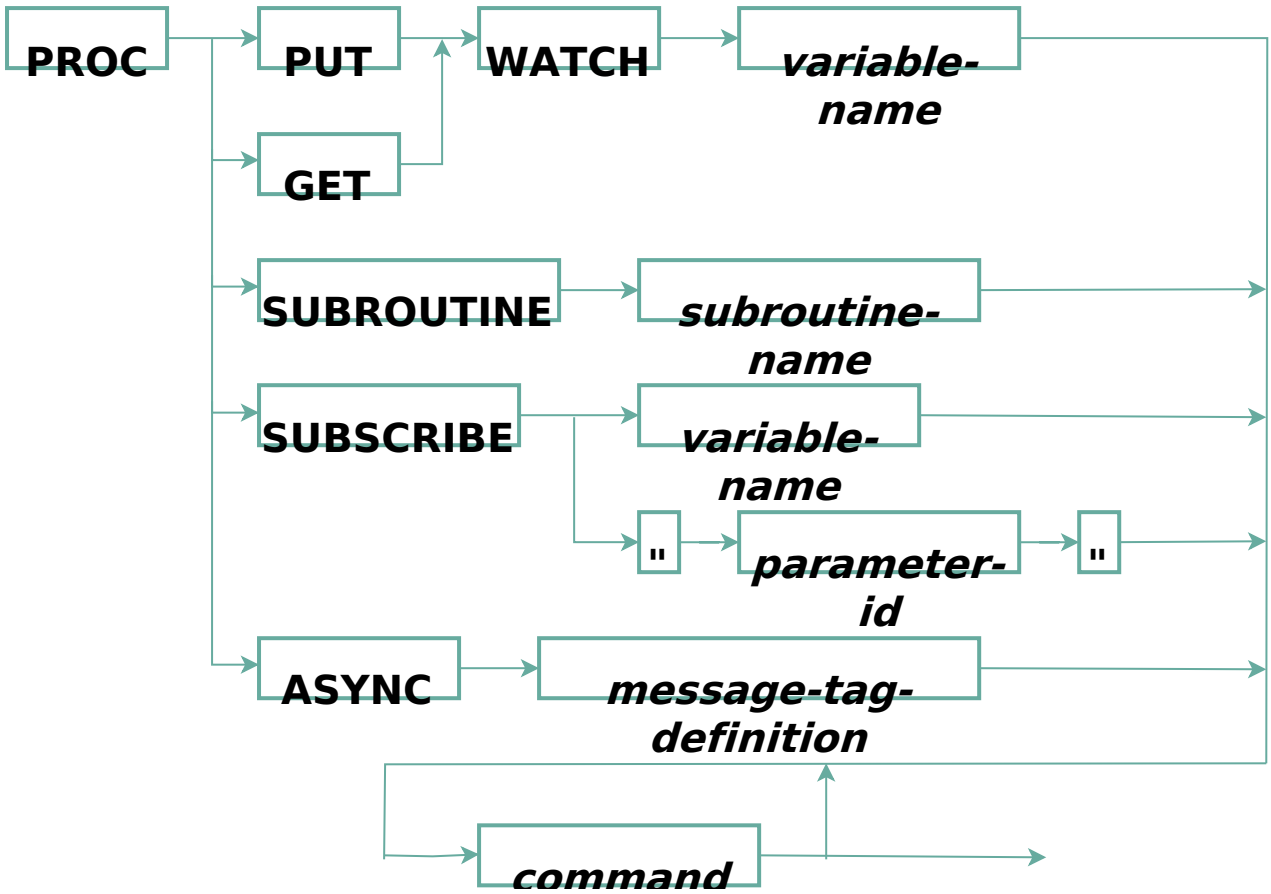
to the other parts of the software, does the device driver procedure implement the interface to the device itself. Based on the communication protocol selected in the equipment setup (the protocol handles the low level coding of the data exchanged with the device), procedures handle the chats done between the MNC and the device to perform parameter settings or to gain some information from the device.

The description of the PROC statement in the following chapter together with the examples supplied with this manual explain the usage of procedures. Generally the following rules apply to device driver procedures:

- Procedures are executed by the driver in the same order as they are defined in the driver specification.
- The driver knows GET-type procedures which read some information from the device into one or more variables and PUT-type procedures which send settings to the device.
- A procedure may handle one or more parameters.
- The driver skips any procedures where actually nothing is to do. For example a PUT-type procedure is skipped unless at least one of the values handled by it has been changed by the operator.
- A variable which has it's data source in the device must be handled at exactly one GET-type procedure.
- A variable being a device setting must be handled at exactly one PUT-type procedure.

7.2.4.1 The PROC statement

Driver procedures are defined with the PROC statement. The PROC keyword is followed by one of `PUT`, `GET`, `SUBSCRIBE`, `SUBROUTINE` or `ASYNC`. Chapter Using subroutines tells more about this type of procedures. `ASYNC` procedures handle messages sent by the device in an unsolicited way. Chapter ASYNC procedures. Procedures do not have an explicit*end* keyword, a procedure automatically ends where another definition (procedure, variable or table) starts.



`PUT` and `GET` procedures must have a `WATCH` keyword followed by one or more names of variables this procedure shall be bound to.

Generally, any variable which is referenced in a procedure must be defined before using the `VAR` statement. Although the driver language allows to define variables between procedures (a `VAR` statement closes an open procedure definition), the drivers supplied by SatService GmbH first define all variables and then all procedures for this reason.

Examples

There are two complete device drivers listed at the end of this section, showing several flavors of procedure definitions:

- NDSatCom-KuBand-Upconverter device driver
- Tandberg-Alteia device driver

Variable subscriptions

The `PROC SUBSCRIBE` variant of procedure definitions permits to react on changes of parameters controlled by other devices drivers. Such a procedure is called every time the referenced parameter changes. The parameter to listen to may be defined directly with its full parameter name in double quotes (e.g. `DEVICE.some.name`) or by the name of a variable in this device driver which contains the name of the parameter to listen to.

SUBSCRIBE type procedures are executed in the thread context where the parameter it's listening to is changed, not in the thread context of the local device driver. For this reason, `SUBSCRIBE` procedures never should contain IO-related statements like `PRINT`, `INPUT`, `READ`, oder `WRITE`. Communicating to the device from different program threads may break the communication protocol. That is much similar like when two people are talking at the same time, probably you won't understand any of them.

7.2.4.2 ASYNC procedures

Device drivers may define `ASYNC` procedures which handle messages sent by a device without a foregone request sent by the MNC. While the ***sat-nms*** device driver architecture is targeted to devices which only reply if requested to do this, `ASYNC` procedures provide a ways to handle unsolicited messages.

Basic concept

The concept behind `ASYN` procedures is a communication protocol which listens to the device all the time, reads all messages coming from the device. Each message is forwarded to the `ASYNC` procedures defined to check which procedure is in charge for this particular message.

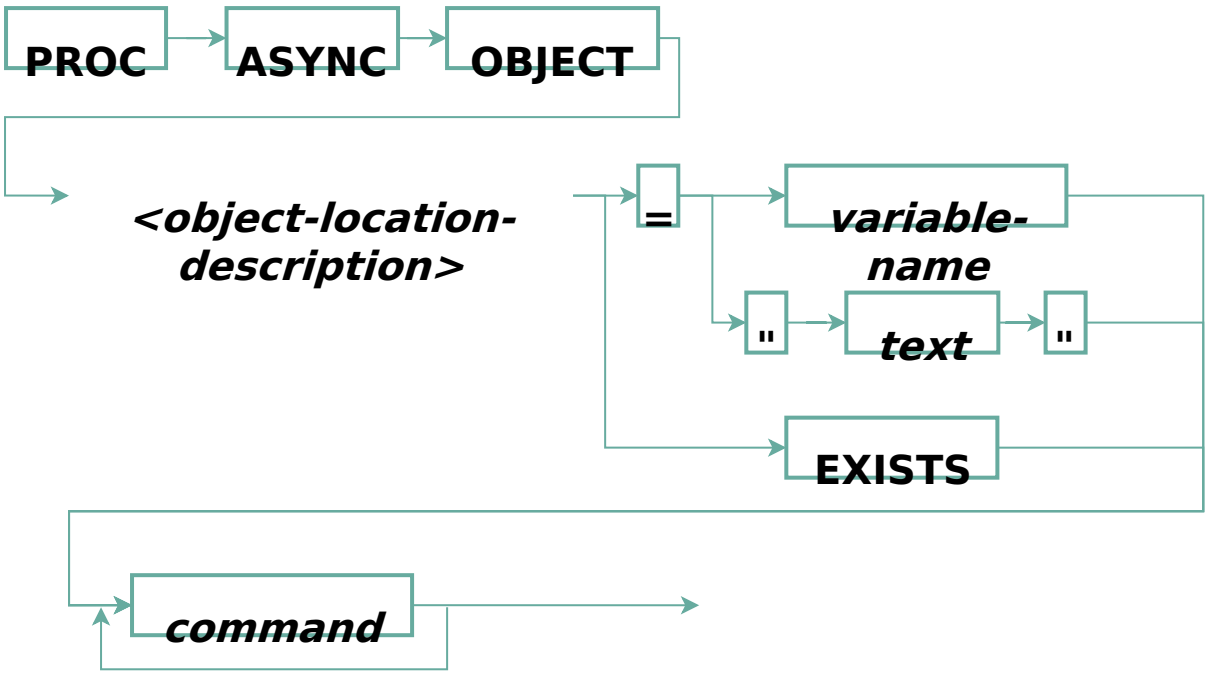
The check ist done by comparing so called message tags to a reference value defined for the procedure. For JSON based protocols the message tag is one value in the JSON document.

There are a number of constraints you have to consider if a device sends messages without being requested to, specially if the device responds to requests *and* sends unsolicited messages as well:

1. Reading data from a device asynchronously requires a ***sat-nms*** communication protocol definition that supports this. The protocol must listen to the device all the time and forward any incoming message to the device driver. If the protocol does not support this, no `ASYNC` procedures will be executed in a driver, even if they are defined.
2. You can no longer assume, that reading a device's reply after sending a request contains the expected data. You have to design the driver in a way, that all commands and status requests are sent in PUT procedures without reading the device's reply. All replies ans unsolicited messages have to be read in `ASYNC` procedures.
3. There are devices which provide separate communication channels for replies to requests and for asynchronous messages. With such a protocol the above restriction does not apply.
4. Commands and requests *never* must be sent from within an `ASYNC` procedure. As the `ASYNC` procedure runs asynchronously to the device driver thread, this may collide with commands sent from a `PUT` procedure.

PROC ASYNC for JSON coded protocols

For JSON based protocols the definition of an asynchronous procedure starts with the keywords `PROC ASYNC OBJECT`, followed by an object path definition as used with the `REST SET` and `REST PARSE` statements. You find an extensive description about the JSON object model and object paths in chapter 2.2.6 REST I/O functions

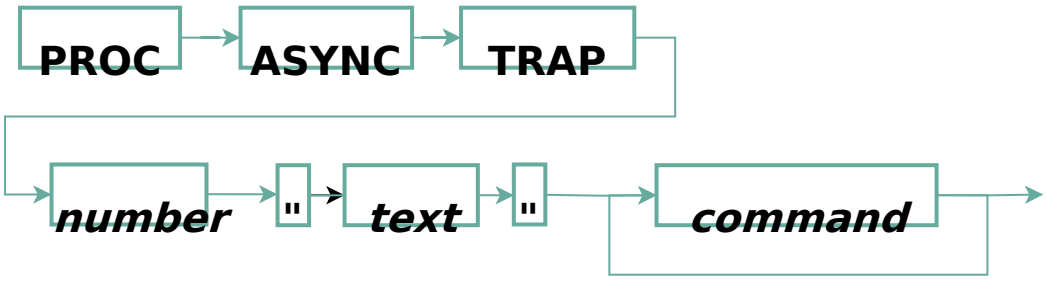


The object path definition is terminated either by the keyword `EXISTS` or a `=` character and the reference value this JSON object gets compared to. The reference value may be a quoted string or the content of a driver variable. The procedure body gets executed if the JSON object defined by the object path contains a value equal to the reference value following the `=` character.

If the keyword `EXISTS` is used, the procedure gets executed if an object at the given location exists in the message, the value assigned to it doesn't care.

PROC ASYNC for SNMP protocols

With SNMP, `PROC ASYNC TRAP` may be used to process traps issued by a device. The syntax for this kind of procedure is as follows:



The keywords `PROC ASYNC TRAP` are followed by the SNMP trap number and a pattern text. The SNMP trap calls this procedure if it has been issued from the IP address set for this device, the SNMP trap number matches the number stated with this procedure and the text submitted by the trap receiver contains the pattern text.

The trap receiver in the M&C software processes every trap it receives and translates it to a multiline string which is passed to the procedure(s) which are in charge for this trap. The first

line of this string contains the trap type, the IP address from which it was issued, the trap number and the specific number. If there are SNMP varbinds with this trap, for each varbind a line follows with the OID and the value if this varbind.

Example:

```
V1TRAP from 127.0.0.1 trapno=6 specific=99
1.3.6.1.4.1.53491.550.2.3.19.0=-76.33
1.3.6.1.4.1.53491.550.2.3.24.0=2
```

The varbinds may be interpreted with `INPUT` statements in the procedure body:

```
PROC ASYNC TRAP 6 "53491.550.2.3.19.0"
   INPUT "53491.550.2.3.19.0=" TRM 10 inputLevel
   INPUT "53491.550.2.3.24.0=" TRM 10 XLT tLock inputLock
```

The procedure above gets called for a trap (6) containing a varbind ending on "53491.550.2.3.19.0". The value of this varbind gets assigned to the driver variable *inputLevel*. The varbind ending on "53491.550.2.3.24.0" is assigned to 'inputLock' after translating it from the SNMP number encoding to the value expected in the driver. Using OID fragements starting with the deivce's enterprise ID is a reliable way to identify varbind OIDs in the trap text.

Remarks

- The trap number in the procedure header may be set to*to make the procedure accept any trap number. For SNMPv3 traps / informs the trap number must be *as SNMPv3 traps do not contain trap numbers.
- The pattern text in the procedure header may be set to "*" to accept any trap without checking if it contains the pattern text.
- `PROC ASYNC TRAP` procedures require the M&C server's trap receiver to setup properly. The file 'traprec.json' configures the properties of the trap receiver, the M&C user manual gives a conprehensive description of this configuration file.

7.2.5 Basic I/O functions

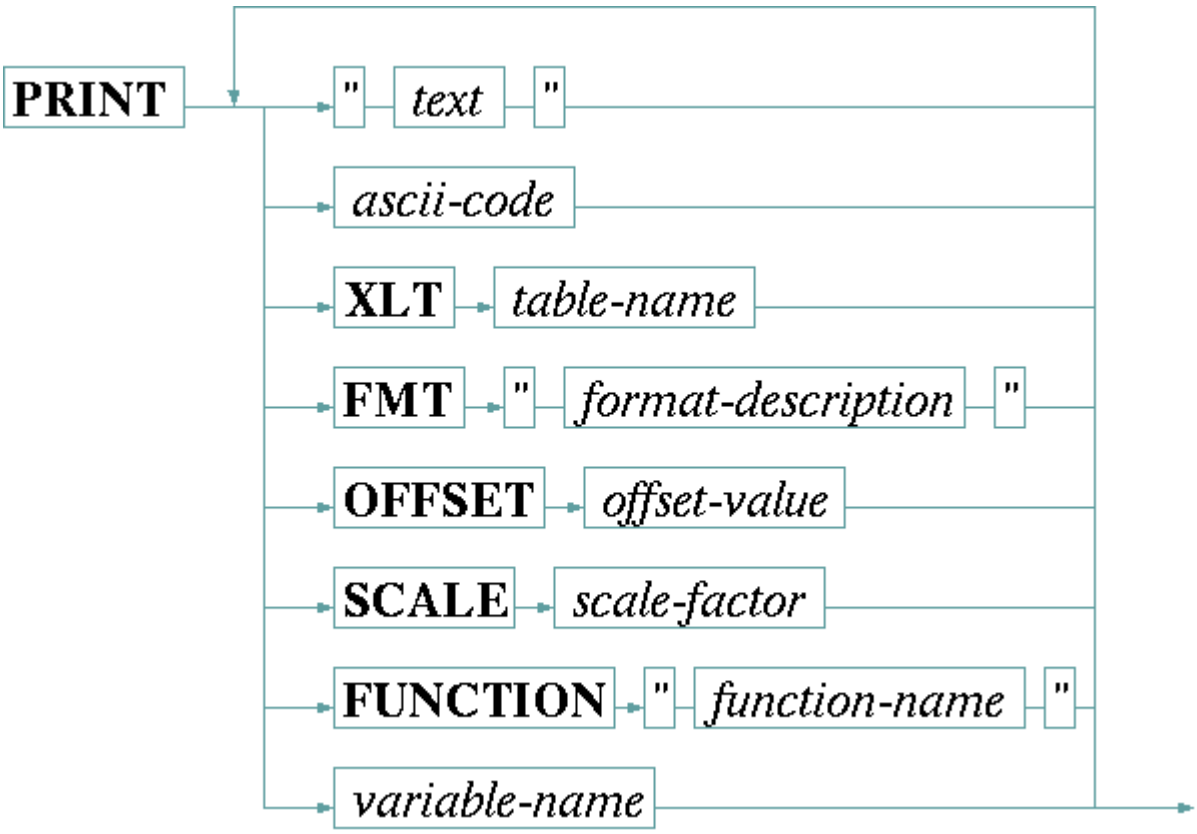
The primary function of a driver procedure is to acquire some information from the device (`GET` procedure) or to set one or more parameters from the data commanded somewhere else in the software (`PUT` procedure). The ***sat-nms*** device driver definition language provides four highly flexible statements for this purpose. They are:

PRINT	Composes/formats s (readable) string from multiple elements which may be text fragments, numbers (formatted in various ways), tokens from a `CHOICE` variable and more. The composed string is packed in a protocol frame and sent to the device. `PRINT` is used with devices which implement a text based protocol.

INPUT	The INPUT statement complements the PRINT statement. It reads a message from the device, strips off the protocol frame parses the received data according to the rules specified with the INPUT statement.
WRITE	The WRITE statement is the equivalent to PRINT for devices providing a binary communication protocol. WRITE sets up a binary data structure from constant and variable values, packs this structure in a protocol frame and sends it to the device.
READ	Reading data from a device providing a binary communication protocol is done with the READ statement. As the counterpart of the WRITE statement it reads a message from the device, strips off the protocol frame and provides means to interpret the received data as a binary structure variables.

7.2.5.1 The PRINT statement

The PRINT statement is used to send commands to a device which uses a text based protocol. It composes a command string from the elements following the the PRINT keyword, packs this string in the protocol frame defined by the communication protocol and sends this message to the device. The syntax of the PRINT statement is:



The table below describes the elements which may follow the PRINT keyword.

"text"	Plain text, enclosed in double quotes, is added to the output buffer as it is.

ascii-code	A decimal number is interpreted as the ASCII code of a single character to output. You may use this to send special characters like carriage-return or line-feed to the device.
XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the next variable value which shall be printed through this translation table. Chapter Using conversion tables gives more information about tables in general.
FMT "..."	The `FMT` keyword, followed by a format description in double quotes, tells the device driver to format the next variable following the format description given. `FMT` is used to format numeric variables into the representation the device expects in it's remote control command. `INTEGER` variables may be printed using a floating point format, and `FLOAT` variables may be printed with a `d` or `x` format likewise. With `FLOAT` variables you should format in any case as internal precision/rounding problems may cause unpredictable results when floating point values are printed unformatted.

A format description consists of the following elements:

format characters	description
d b x X f	The first letter of the format description defines the general number representation: `d` (decimal), `b` (binary), `x` (hex, lower case), `X` (hex, upper case) or `f` (floating point)
+	If the `+` option is given, the number is preceded by a +/- character even if it is positive. Together with the `0` option below, the sign appears as an additional character at the first column, hence the field width is increased by one on this case.
0	If the `0` option is given, the field is padded up with zeroes instead of spaces.
1 .. 99	Here follows a one or two digit field width. The field width is the total number of characters the formatted number occupies including the padding characters. If there are more digits needed to show a number correctly, the field with is enlarged automatically. Specifying a field width `1` disables the right orientation in a fixed width completely. This specially is useful for floating point numbers where only the number of fraction digits shall be fixed, not the complete field width.
.	For floating point formats the dot separates the precision from the field width.

format characters	description
0 .. 9	The dot is followed by a one digit specification of the number of fraction digits which shall be printed. The dot and fraction digits specification is valid only with the floating point format.

The table below describes the more elements which may follow the `PRINT` keyword.

OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, tells the driver to add the offset value to the next variable before it is printed.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, tells the driver to multiply the next variable with the scale factor before it is printed
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
variable-name	A variable name tells the driver to print the value stored in this variable. Usually the *commanded* value is used rather than the value which has been read back from the device. If, however, there has been never a value commanded, or if this variable is a read only variable, the value recently read from the device is used.

Before the variable is printed to the output buffer, any `XLT`, `FMT`, `OFFSET` or `SCALE` operations which have been specified are applied. This happens in a fixed order:

1. If a `SCALE` operation has been specified, this is done first.
2. The `SCALE` operation is followed by an `OFFSET` addition.
3. The so scaled value is processed through a `FUNCTION` if there is one.
4. The processed value is formatted according to a `FMT` specification if one is given.
5. Finally, the value gets translated through a translation table, if an `XLT` operation has been specified.

This scheme applies to numeric (`INTEGER`, `HEX`, `FLOAT`) and to text type variables as well. As soon as a `SCALE`, `OFFSET`, `FUNCTION` or `FMT` keyword is present in front of a variable, this gets converted to a floating point number. Strings which cannot be converted give a zero value.

Example

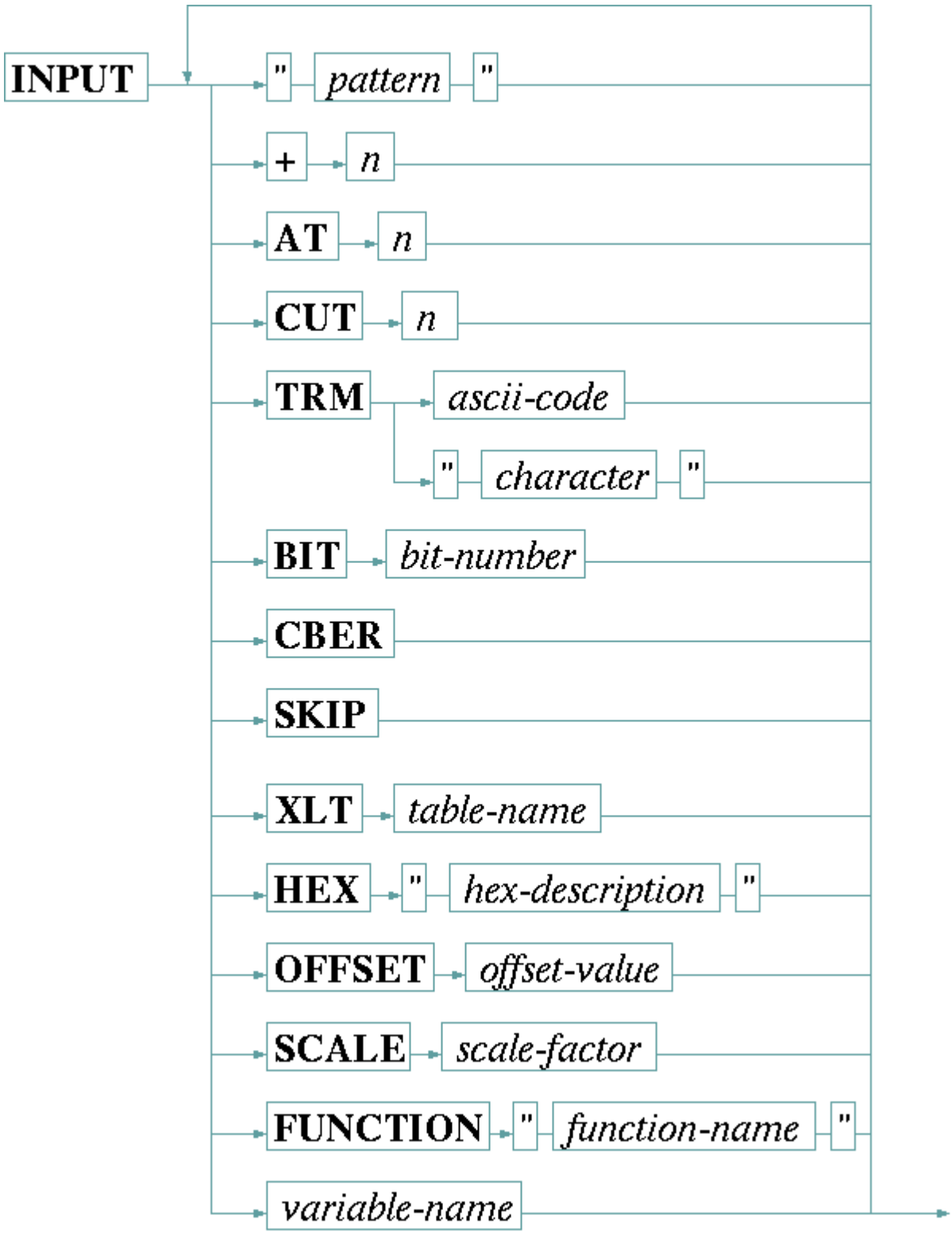
```
PRINT "TUN;FRQ=" SCALE 8.0 FMT "d06" frequency
```

The example above sends a command of the format `TUN;FRQ=######` to the device. The

`FLOAT` variable `frequency` is multiplied by `8.0` and the output as an integer number, 6 digits wide with leading zeroes.

7.2.5.2 The INPUT statement

The `INPUT` statement is used to read and interpret the reply of a device which uses a text based communication protocol. It reads a protocol frame from the device and parses the received message according to the rules defined by the description following the `INPUT` keyword. The syntax of the `INPUT` statement is:



While parsing the device's reply, the driver deals with three information buffers.

1. *original reply*
2. *pad* buffer
3. *value* buffer

To understand the way how the parsing operations work which may follow the `INPUT` keyword, it is necessary to learn how the driver uses these buffers.

- The first buffer contains the original reply string as it is received from the device (with the protocol frame stripped off).
- The driver initially copies this string into a second buffer called *pad*. Search operations (text in double quotes) and the "+" operation modify the *pad*. The AT operation is the only way to restore the *pad* from the original data.
- Every time, the *pad* is modified, the driver copies it's contents into the third buffer called *value*. *Value* is the string which will be assigned to a variable if it appears in the list of arguments to the `INPUT` statement. Operations in the `INPUT` statement like `CUT`, `TRM`, or `XLT` work on the *value* buffer but leave the *pad* unchanged. This is important to understand if you have to parse the value of more than one variable in a single `INPUT` statement.

" pattern "	Searches the text/pattern in the current *pad* buffer. If the pattern is found, all text including the first occurrence of the pattern is removed from the beginning of the *pad* buffer. The pattern text may contain escape sequences like `\n` or `\t` to search for line feeds or tabulator characters.
+ n	Removed `n` characters from the beginning of the *pad* buffer.
AT n	Restores the *pad* buffer with a substring of the original data, starting at column `n`.
CUT n	Cuts the *value* buffer to a length of `n` characters.
TRM "c"	Removes the first occurrence of the termination character and all following text from the *value* buffer. The termination character may be specified as a decimal number (e.g. 10 is the ASCII code for line feed) or as a single character enclosed in double quotes.
BIT n	Interprets the *value* buffer as a decimal number and isolates the bit number `n` (`n = 0` means the least significant bit). The value buffer is replaced by `0` or `1` depending on the bit value.
CBER	Interprets the *value* buffer as a bit error rate value as it is returned by Comstream modems. A string `mn` is converted into the more common notation `mE-n`.
SKIP	Removes any whitespace (space characters, line feeds, carriage returns or tabs) from the beginning of the *value* buffer.
XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the *value* buffer through this translation table from right to left. Chapter Using conversion tables gives more information about tables in general.

OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, interprets the *value* buffer as a floating point number and replaces the buffer with the sum of the value and the offset.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, interprets the *value* buffer as a floating point number and replaces the buffer with the product of the value and the scale factor.
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
variable-name	A variable name tells the driver to assign the contents of the*value* buffer to this variable. The driver performs format conversions and range checks according to the type and limit specifications made in the VAR definition of this variable.
HEX ".."	Interprets the contents of the *value* buffer as a hex number and replaces it by it's decimal equivalent.

The format definition string which must follow the HEX keyword may contain the following characters:

L	The hex number is in *little endian* notation. This means the least significant byte is written at first. If no `L` character is in the format string, *big endian* notation is assumed.
S	The hex number is signed. The number of digits is used to calculate the sign extension: `80` for example is calculated as `-1` as the two digit number is assumed to be a byte value. `0080` in contrast is computed to `128` as a 16 bit number of this value is positive.

Remarks

If a value is not assigned to a variable as you expect, you should check the following issues:

- Numeric values are converted from string to numeric representation in a very lenient way. For instance, any leading non-numeric characters are ignored. If a numeric variable is not assigned as expected, in most cases there is a range violation. The driver does not accept values outside the range defined for this variable.
- `CHOICE` variables and tables (referenced with the XLT operation) expect an input value which *exactly* matches one of the valid choices. See chapterUsing conversion tables to read how you can make tables somewhat fault tolerant.

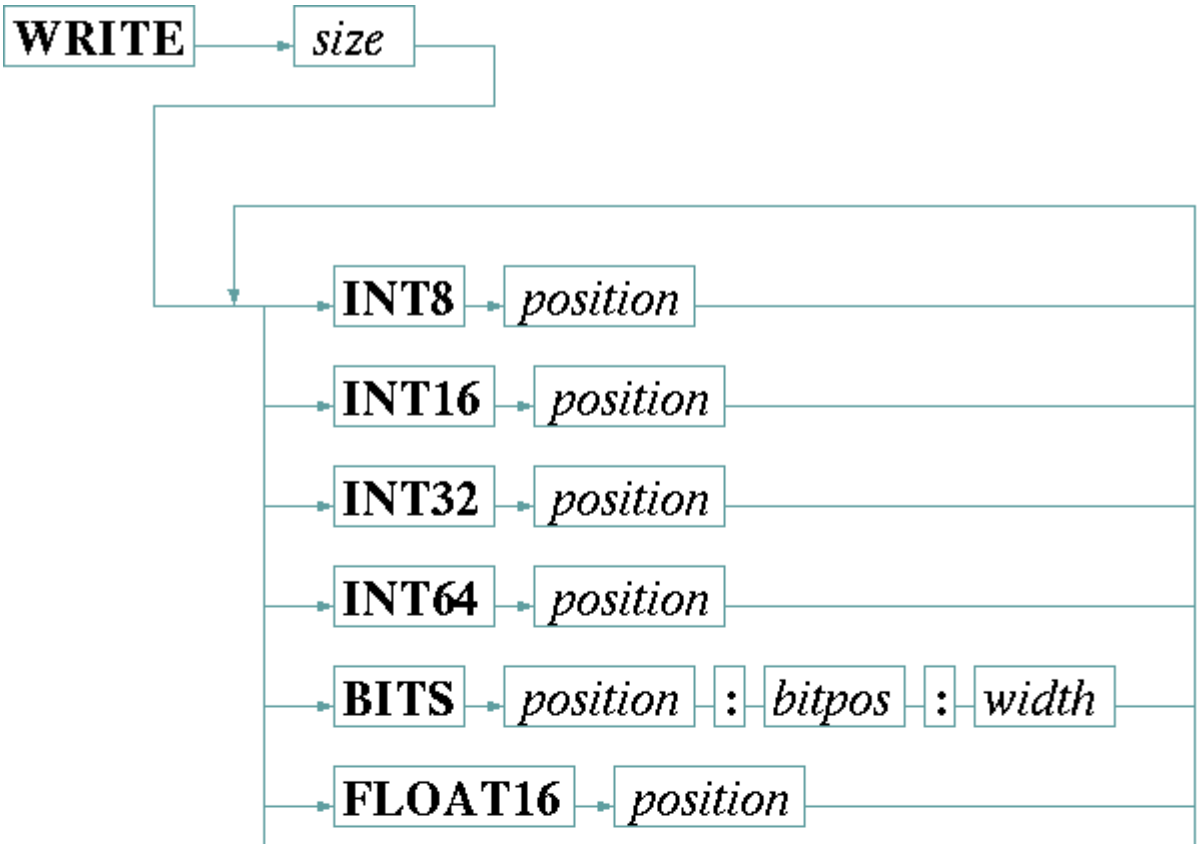
Example

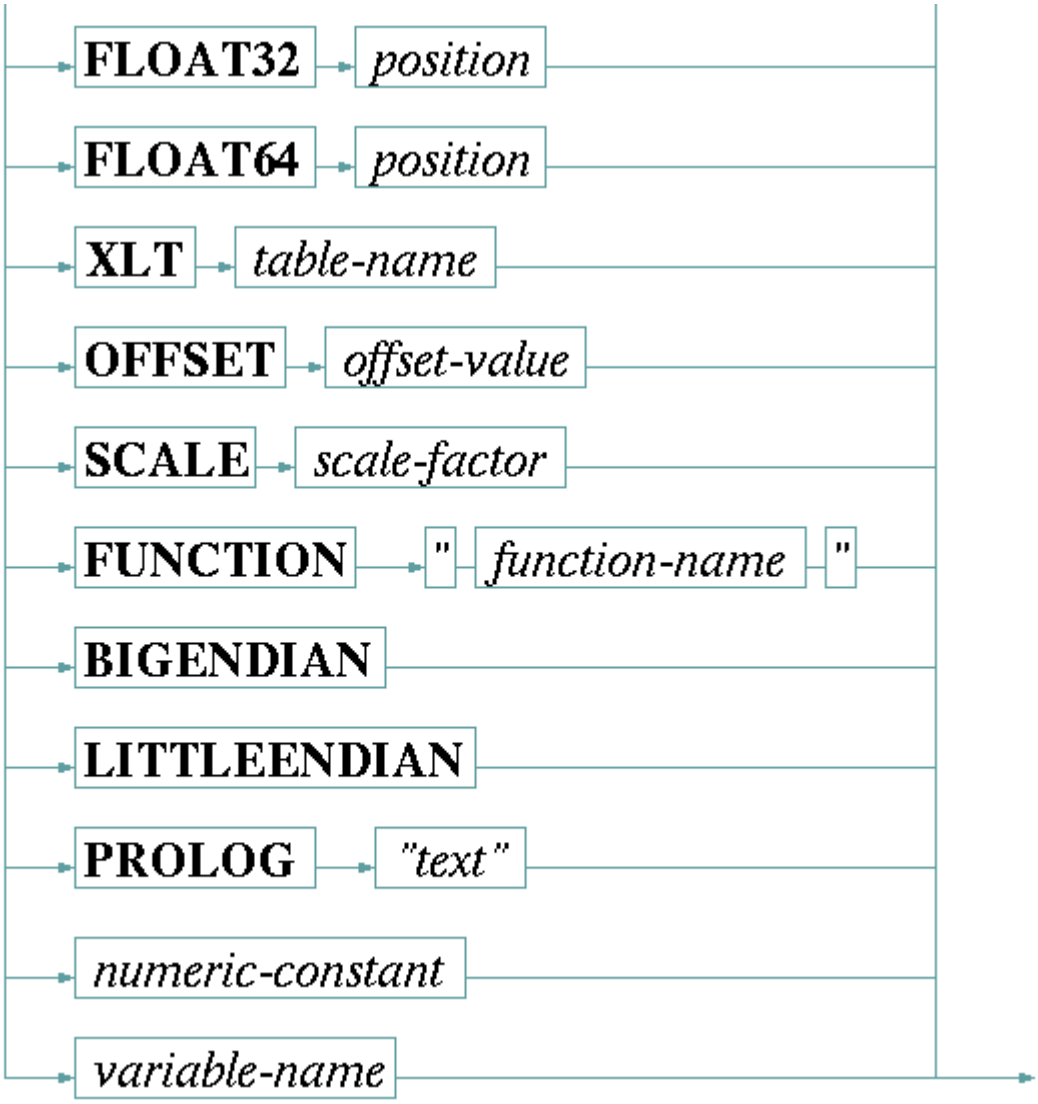
```
INPUT AT 2  SCALE 0.001          tx.frequency
      AT 11 SCALE -0.1 OFFSET 30.0 tx.gain
      AT 15 CUT 1 XLT T03        faults.01
      AT 17 CUT 1 XLT T02        info.if
      AT 19 CUT 1 XLT T01        tx.on
      AT 21 CUT 1                faults.02
      AT 22 CUT 1                faults.03
      AT 23 CUT 1                faults.04
      AT 24 CUT 1                faults.05
      AT 25 CUT 1                faults.06
      AT 26 CUT 1                faults.07
```

The example above - taken from the NDSatCom-KuBand-Upconverter device driver - parses all settings an faults flags from a single status string the upconverter returns. Note that the numeric variables `tx.frequency` and `tx.gain` are isolated from the string without specifying a length or a termination character. The number recognition routine automatically stops where it finds a non-numeric character after the number to read. The other values however all are cut to one character length before they are interpreted.

7.2.5.3 The WRITE statement

The `WRITE` statement is used to send data to the device which uses a binary communication protocol. Such a device expects a command as a binary data structure rather than as a readable text. The syntax of the `WRITE` command is:





The `WRITE` statement creates a buffer of the specified number of bytes which initially is filled with zeroes. Variables or constant values are copied into this buffer according to the size and position specifications given.

The `WRITE` statement by default works in *little endian* mode. 16, 32 or 64 bit values are copied into the buffer with the least significant byte first. Specifying `BIGENDIAN` mode reverses this byte order.

size	The size (number of bytes) of the message buffer to create must be specified as the first argument following the `WRITE` keyword. The size value is expected as a decimal number.
INT8 p	The `INT8` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a single byte value at the given position (position zero denotes the first byte in the buffer).

INT16 p	The `INT16` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a two-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless `BIGENDIAN` was specified.
INT32 p	The `INT32` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a four-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless `BIGENDIAN` was specified.
INT64 p	The `INT64` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a eight-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless `BIGENDIAN` was specified.
BITS p : b : w	The `BITS` keyword tells the driver to place the next value as a 1 to 7 bit wide bit field within a certain byte. In this case the position parameter consists of three values, separated by colons:
	position `p` addresses the byte within the buffer.
	bitpos `b` defines a number of bit positions the value shall be shifted left before it is pasted into the byte.
	width `w` the number of bits (starting with the least significant one) which shall be copied from the next constant or variable into the destination byte.
FLOAT16 p	The `FLOAT16` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a half precision (16 bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless BIGENDIAN was specified.
FLOAT32 p	The `FLOAT32` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a single precision (32 bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless BIGENDIAN was specified.

FLOAT64 p	The `FLOAT64` keyword, followed by a decimal position value, tells the driver that the next constant or variable shall be copied to the buffer as a double precision (64-bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is placed first unless BIGENDIAN was specified.
XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the next variable value which shall be printed through this translation table. When output by a `WRITE` statement, any non-numeric variable must be translated through a table into a numeric value. Chapter Using conversion tables gives more information about tables in general.
OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, tells the driver to add the offset value to the next variable before it is pasted into the buffer.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, tells the driver to multiply the next variable with the scale factor before it is pasted into the buffer.
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
BIGENDIAN	The `BIGENDIAN` keyword turns the byte order from *little endian* (LSB first) which is the default to *big endian* (MSB first). This applies to all values written after BIGENDIAN.
LITTLEENDIAN	The `LITTLEENDIAN` keyword turns the byte order back to *little endian* (LSB first). This applies to all values written after `LITTLEENDIAN`.
PROLOG "text"	The `PROLOG` keyword followed by a quoted string sets this string as a *prolog* to be sent before the binary data specified in the WRITE statement. Actually, this is only usable together with the HTTP11 protocol to generate a HTTP 1.1 POST request. Example: `PROLOG "POST some/url "` prepends this text before the binary data.
numeric-constant	Numeric (decimal) constants can be used like variable names. The constant value is pasted into the output buffer in this case. If a quoted string is given, this is treated as a hexadecimal constant and translated into a numeric value.

variable-name	A variable name tells the driver to paste the value stored in this variable into the output buffer according to the size and position specification given before. Usually the *commanded* value is used rather than the value which has been read back from the device. If, however, there has been never a value commanded, or if this variable is a read only variable, the value recently read from the device is used.

Before the variable is printed to the output buffer, any `XLT` , OFFSET `or` SCALE` operations which have been specified are applied. This happens in a fixed order:

1. If a `SCALE` operation has been specified, this is done first.
2. The `SCALE` operation is followed by an `OFFSET` addition.
3. The so scaled value is processed through a `FUNCTION` if there is one.
4. The processed value is formatted according to a `FMT` specification if one is given.
5. Finally, the value gets translated through a translation table, if an `XLT` operation has been specified.

Remarks

- Each variable name or constant must be preceded by an `INT8` , `INT16` , `INT32` , `INT64` or `BITS` specification to be pasted properly into the output buffer.
- Only `INTEGER` , `HEX` or `FLOAT` type variables can be used directly with the `WRITE` statement. For other variable types an `XLT` table translation must be used to turn the variable into a numeric value.
- `FLOAT` variables are converted to 64-bit integer values before they are used. `OFFSET` or `SCALE` operations are done before this conversion.

Example

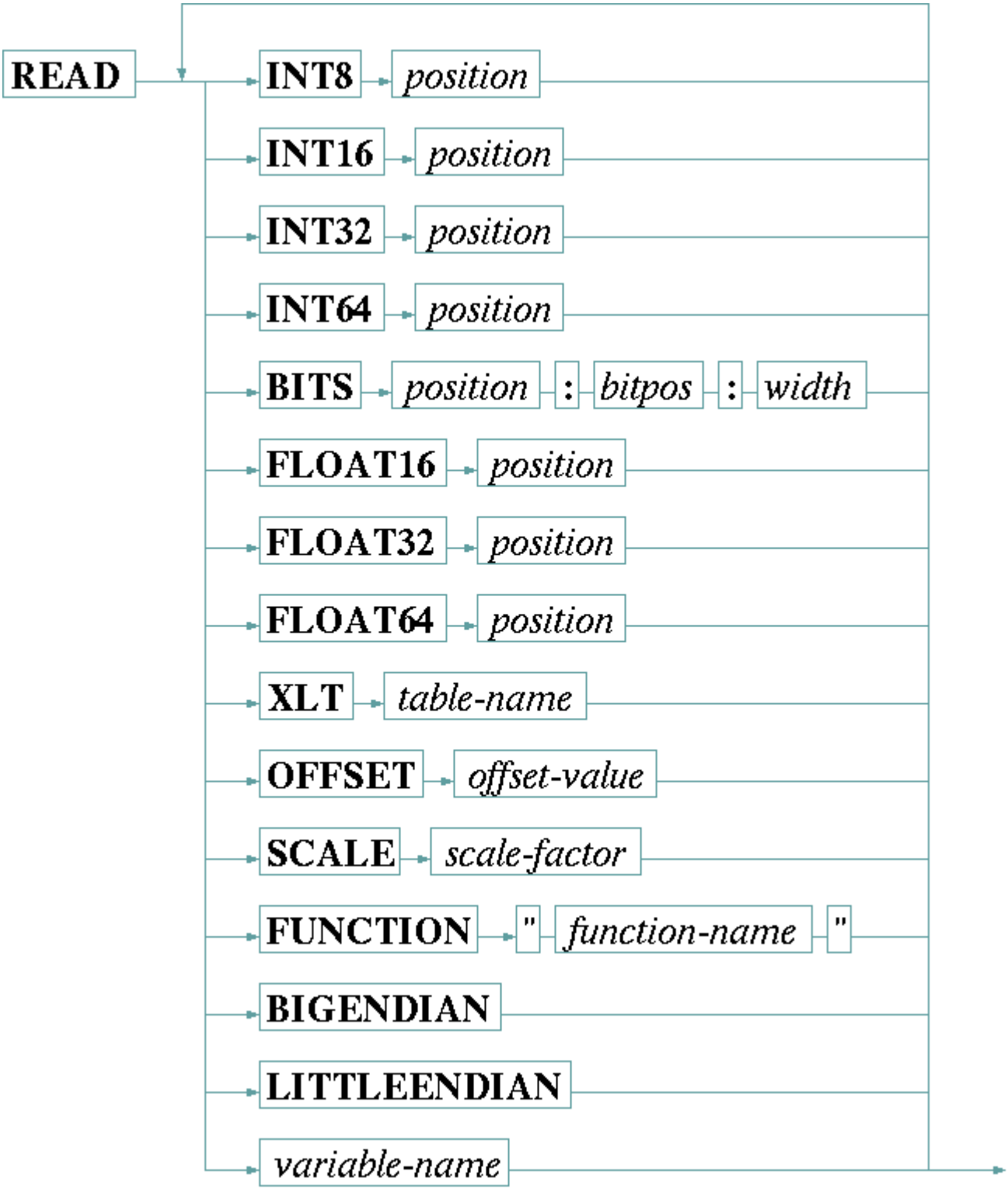
```
TABLE T02 "QPSK=0,BPSK=1"

...

WRITE 3 BIGENDIAN
   INT16 0  9734
   INT8  2  XLT T02        tx.mod.type
```

The example above -- taken from the Radyne DVB3030 modulator driver -- explains how to write a `CHOICE` parameter to a device. Table `T02` is used to convert the QPSK/BPSK selection into the numeric values 0/1. The WRITE statement sends a message (three bytes long plus protocol frame overhead) in *big endian* byte order. The first two bytes are filled with the decimal constant `9734` (a command code). The third byte is set to `0` or `1` according to the contents of `tx.mod.type` .

7.2.5.4 The READ statement

Receiving and interpreting of data returned from devices which use a binary communication

protocol is done with the `READ` statement. The `READ` statement gets a message from the device, strips off the protocol frame and places the result in an internal buffer. Subcommands/operations following the `READ` keyword are used to copy values from the buffer into one or more variables of the device driver.



The `READ` statement by default works in 'little endian' mode. 16, 32 or 64 bit values are read

from the buffer with the least significant byte first. Specifying `BIGENDIAN` mode reverses this byte order.

INT8 p	The `INT8` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a single byte value at the given position (position zero denotes the first byte in the buffer).
INT16 p	The `INT16` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a two-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless `BIGENDIAN` was specified.
INT32 p	The `INT32` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a four-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless `BIGENDIAN` was specified.
INT64 p	The `INT64` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a eight-byte value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless `BIGENDIAN` was specified.
BITS p : b : w	The `BITS` keyword tells the driver to read a 1 to 7 bit wide bit field from a certain byte in the buffer. In this case the position parameter consists of three values, separated by colons:
	position `p` addresses the byte within the buffer,
	bitpos `b` defines the bit-number (0 to 7) of the least significant bit to read.
	width `w` the number of bits which shall be copied from the buffer into the destination variable.
FLOAT16 p	The `FLOAT16` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a half precision (16 bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless BIGENDIAN was specified.

FLOAT32 p	The `FLOAT32` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a single precision (32 bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless BIGENDIAN was specified.
FLOAT64 p	The `FLOAT64` keyword, followed by a decimal position value, tells the driver that the next variable appearing in the argument list shall be filled from a double precision (32 bit) IEEE754 floating point value starting at the given position (position zero denotes the first byte in the buffer). The least significant byte of the number is read first unless BIGENDIAN was specified.
XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the value read through this translation table from right to left. Chapter Using conversion tables gives more information about tables in general.
OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, tells the driver to add the given offset to the value read.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, tells the driver to multiply value read with the given factor.
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
BIGENDIAN	The `BIGENDIAN` keyword turns the byte order from *little endian* (LSB first) which is the default to *big endian* (MSB first). This applies to all values read after BIGENDIAN.
LITTLEENDIAN	The `LITTLEENDIAN` keyword turns the byte order back to 'little endian' (LSB first). This applies to all values read after `LITTLEENDIAN`.
variable-name	A variable name tells the driver to store the value read into the variable.

Example

```
READ BIGENDIAN
   INT32 1  SCALE 0.000001   tx.frequency
   INT32 7                   tx.mod.dataRate
   INT32 11 SCALE 0.000001   refClkFreq
   INT8  15 XLT T01          refClkSrc
   INT8  16 XLT T02          tx.mod.type
   INT8  17 XLT T03          tx.mod.fec
   INT16 22 SCALE 0.1          tx.power
   INT8  24 XLT T04          tx.on
   INT8  24 XLT T04          internal.tx.on
   INT8  25 XLT T05          tx.mod.cwMode
   INT8  26 XLT T06          tx.mod.spectrumInvert
   INT8  28 XLT T14          tx.ifc.hardware
   INT8  29 XLT T06          tx.ifc.clockPhase
   INT8  30 XLT T06          tx.ifc.dataPhase
   INT8  31 XLT T07          tx.mod.clockSource
   INT8  44 XLT T08          info.maskEnable
   INT32 45                  info.alarmMask
   INT32 49                  tx.mod.symbolRate
   INT8  53 XLT T09          tx.ifc.framingMode
   INT8  54 XLT T10          tx.mod.rollOff
   INT8  55 XLT T11          config.control
   INT8  57 XLT T12          modemType
```

The example above - taken from the Radyne DVB3030 modulator driver - shows the main settings `READ` statement of this driver. In one step almost all parameter settings are read.

7.2.5.5 Function tables in I/O functions

The I/O functions of the ***sat-nms*** driver language (`INPUT` , `PRINT` , `READ` , `WRITE` and the REST I/O functions) all provide with the `FUNCTION` keyword a feature to translate a numeric value along a table of x/y value pairs. The result of the function is calculated by linear interpolation in this table.

File format

A `FUNCTION` requires the table of x/y values to be stored as a simple text file which looks like the following example:

```
# function table, defines a square root function

0.0   0.0
1.0   1.0
4.0   2.0
9.0   3.0
16.0   4.0
25.0   5.0
```

The table consists of two or more lines, each consisting of the x-value followed by one or more space characters and the corresponding y-value. Both values must be valid floating point numbers. Blanc lines and lines starting with a `#` character are ignored.

The x-values need no to be e equally spaced, but the values must be sorted in ascending order, starting with the lowest one.

Calculation method

The function calculates the the y-value to a given x-value by linear interpolation. This means, the curve defined by the x/y pairs is a polygonal line. The table should contain enough x/y-pairs to meet the expected accuracy of the function.

For x-values outside the range of x/y pairs defined in the table, the function extrapolates the y-value using the slope given by the outmost two x/y pairs. You may want to set two additional x/y-pairs outside the real table to set this slope to a particular value.

File name processing

When executed in an I/O statement, the function extends the function name defined in the driver to the name of the file containing the function table. First the letters `.txt` are appended to the function name. The `.txt` file extension makes it easy to open the file with a text editor from the file manager of any operating system. Function table must be placed in the base directory of the ***sat-nms*** installation (usually `/home/satnms/`).

If the function name starts with a `-` character, the name of the device is prepended to the function name. For example, a function name `"-caltable"` would expand to a file name `DEVNAME-caltable.txt` where `DEVNAME` is the name of the actual device. This is useful for e.g. calibration tables which are different for each particular device.

Error handling

The function checks if the file for this function exists. The driver raises a communcation fault if a `FUNCTION` clause with non-existing file is processed. If the file itself cannot be parsed for some reason, the function tries to do its best, recovering from this situation if possible. A note is written to the `.panic.log` file, if such an error occurs.

7.2.6 REST I/O functions

Beside the basic I/O functions described in the previous chapters, the***sat-nms*** driver definition

language provides some statements to handle the communication with devices using a REST API in an efficient way.

The REST API transports settings and status information as structured, JSON-formatted documents over an HTTP protocol. JSON documents are plain text and therefore may be - at least principally - handled by the basic PRINT / INPUT statements. In practice however, parsing a JSON document with the capabilities of INPUT may become difficult and in the end will result in a device driver which is hard to understand and to maintain.

The REST I/O statements contained in the ***sat-nms*** driver definition language take care of the special structure and formatting of JSON documents, helping to focus on the content to send or retrieve rather than the special JSON syntax. The statements for REST communication are:

REST TRANSACT	Sends a REST HTTP request to the device, for HTTP `POST` , `PUT` or `PATCH` with a JSON formatted document body attached. Retrieves the device's reply and parses the document body into an object tree.
REST PARSE	Accesses the objects contained in the most recent reply from the device and assigns the object values to ***sat-nms*** driver variables.
REST SET	Modifies the objects contained in the JSON document or adds new objects to it. Used to prepare a document to be sent to the device with one of the `POST` , `PUT` or `PATCH` HTTP methods.
REST CLEAR	Clears the JSON document in memory. This function is used to prepare a document from scratch for a `POST` or `PATCH` call.
REST SEND	takes the actual JSON document and sends it to the device. It is used with protocols other than HTTP and replaces `REST TRANSACT` there.

Object model

A JSON document is a hierarchically structured collection of key / value pairs. The value in such a pair may be either a constant value, another ancillary collection or an array of these. The object model used in the ***sat-nms*** REST I/O functions flatens this tree-like structure, treats every constant value with its key as an object. The key of this constant value is extended to contain the complete information where in the document this key / value pair is located.

Object paths

This location information can be thought as a path from the document's root down to the definition of the constant value of interest. This path is much like a path in a computer's file system. The higher-order nodes in the document tree are like directories in the file system. For this reason, the ***sat-nms*** driver definition language uses a notation like used with file names to identify an object in the document tree:

```
/settings/modulator/datarate
```

for example denotes an object `datarate` contained in the `modulator` branch of the

document which in turn is expected to be contained in the `settings` branch. The JSON equivalent of this would look like this:

```
{
 "settings": {
  "modulator": {
    "datarate": 2048000;
  };
 };
}
```

As with file names and paths, a path identifying an object may be absolute or relative. The example above uses an absolute path to the object: it starts with a slash character. When you access `/settings/modulator/datarate`, the software makes an implicit *change directory* to `/settings/modulator`. If you access then an object named for example `fec`, the software automatically will expand the object name to `/settings/modulator/fec`. This works inside a single `REST SET` or `REST PARSE` statement, not across separate statements.

Object arrays

The array construct in a JSON document works in a special way and therefore needs some special treatment in the ***sat-nms*** driver definition language. JSON arrays are a collection of equally structured nodes. In most REST APIs, the elements of such a collection are not identified by their location in in the collection but by some unique identifier which appears as a regular key / value pair in each element of the array.

To cope with this construct, the ***sat-nms*** driver definition language exteds the object path definition with "conditions" used to identify a particular collection element:

```
/settings/modulator/output[id=3]/enable
```

addresses the `enable` object in that `output` branch that has a key `id` set to `3`. Conditions may appear after each branch name in the object path. There may be multiple conditions which are combined with a logical **and**:

```
/settings/modulator/connector[number=2][type=OUTPUT]/enable
```

addresses the `enable` object in that `connector` branch that has a key `number` set to `3` **and** `type` set to `OUTPUT`.

Alternatively, array elements may be addressed by their position in the array. The numeric array index starts with `0` for the first array element:

```
/settings/modulator/output[2]/enable
```

addresses the `enable` object in the third (index=2) 'output' branch. Please note, that this only works, if the API of the device to control explicitly permits to access array elements by position.

Furthermore it is not possible to create array elements by position index in the REST SET statement.

Document handling

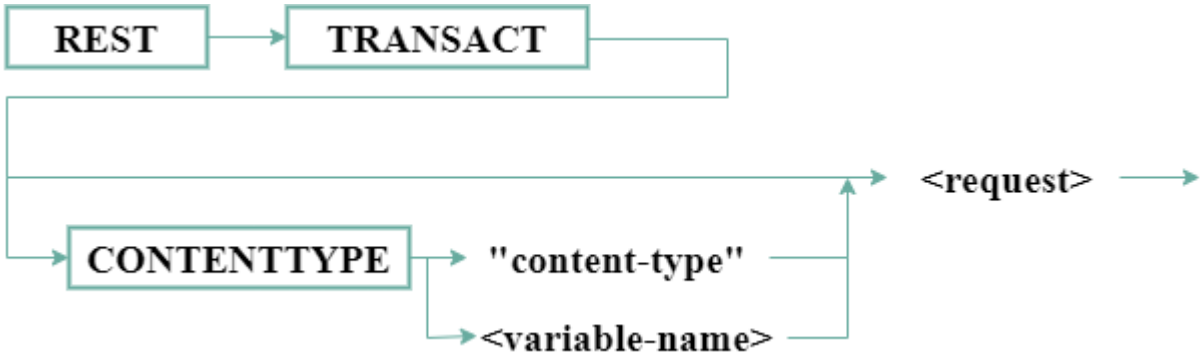
With the REST statements in the ***sat-nms*** driver definition language there is always one JSON document in memory these statements work on. Generally, it is the REST TRANSACT statement which creates this document from the device's reply, REST CLEAR creates an empty document instead. The REST SET statement can be used to add or modify objects in the document, the REST PARSE statement to query objects from there.

With a REST TRANSACT doing a HTTP GET command you can expect to get the information you addressed in the GET command, but POST, PUT or PATCH also may return a document containing some information that might be important for the device driver.

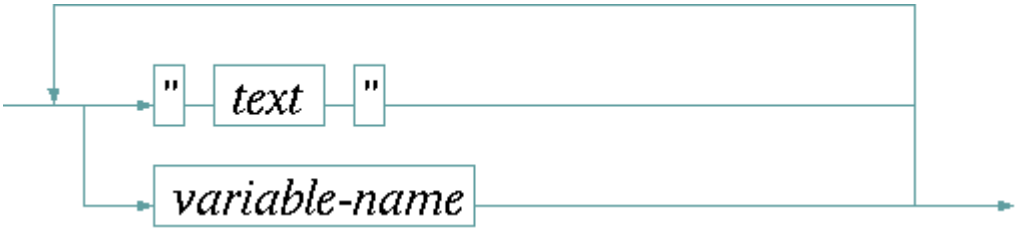
When issuing a POST, PUT or PATCH command, the REST TRANSACT statement sends the existing document to the device using the selected HTTP method, then replaces the document by the parsed reply from the device.

7.2.6.1 The REST TRANSACT statement

The REST TRANSACT statement sends a REST HTTP request to the device, either with or without a JSON formatted document body. It retrieves the device's reply and parses the document body into an object tree.



The request is an arbitrary long sequence of quoted strings and ***sat-nms*** variables. Strings and variable content are concatenated to build the request string:



The request must start with the HTTP method to execute (one of GET, PUT, POST, PATCH or DELETE), followed by one space character and the path of the URL to read. The path may contain URL parameters. Example:

```
GET /api/v1/settings/modulator?slot=2
```

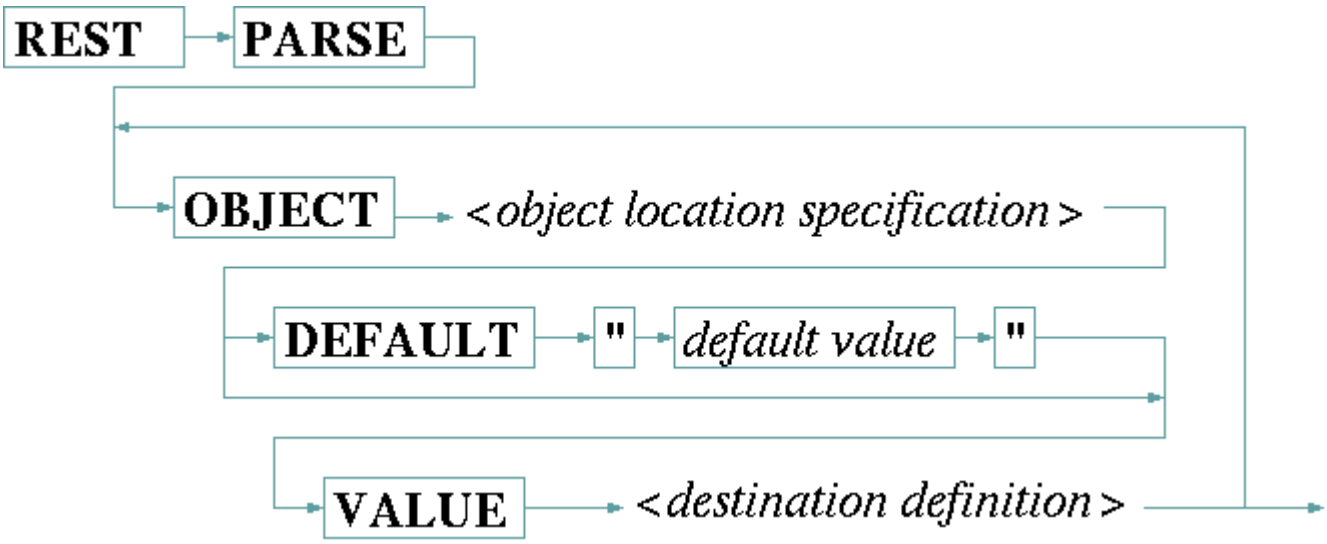
Please note, that no URL-encoding is done on this string, this means that all special character must be escaped (e.g. blanc replaced by '+')

With the POST, PUT or PATCH HTTP methods, REST TRANSACT sends the JSON document actually in memory to the device. This document may be created from scratch or the recent reply from the device may be used as a template which is modified before the transaction is executed. To transact a HTTP method other than GET without sending a JSON document, place a REST CLEAR statement directly above the REST TRANSACT.

REST TRANSACT may be immediately followed by the keyword CONTENTTYPE and a content type definition in double quotes or contained in a variable. This content type definition overrides any PROTOCOLPARAMETER post.content.type=... definition made in the protocol or driver header.

7.2.6.2 The REST PARSE statement

The REST PARSE statement accesses the objects contained in the most recent reply from the device and assigns the object values to ***sat-nms*** driver variables. It uses the same object model as described above for the REST TRANSACT statement to address and extract the desired information from the JSON document the device replied through its REST API.

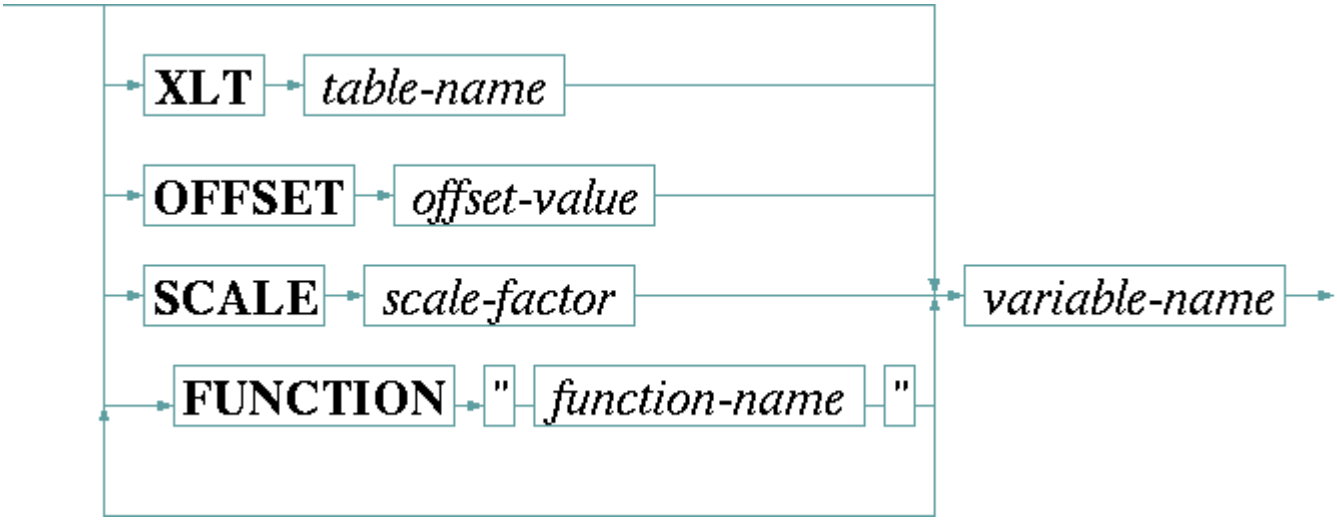


The OBJECT keyword followed by the object location specification addresses the object in the JSON tree to be read. The object location specification works exactly as described above with one difference: non-existent objects are not created but treated as a fault unless there is a default value defined for this object.

The optional DEFAULT keyword denotes the default value to be assigned if the specified object does not exist in the object tree. The default value is given as a quoted string following the DEFAULT keyword. Please note, this value is treated as value read from the JSON document, it undergoes the modifications like XLT, SCALE or OFFSET following VALUE

before it is assigned to the ***sat-nms*** driver variable. If no `DEFAULT` is defined within an `OBJECT` clause and the addressed object is missing in the JSON document, the driver will raise a communication fault.

The keyword `VALUE` starts the definition of the ***sat-nms*** driver variable which shall receive the extracted value together with some modifiers known from the `INPUT` statement. The destination definition in any case ends with then name of the ***sat-nms*** driver variable which may be prepended by zero or more of the modifiers `XLT`, `OFFSET` or `SCALE` as shown below:



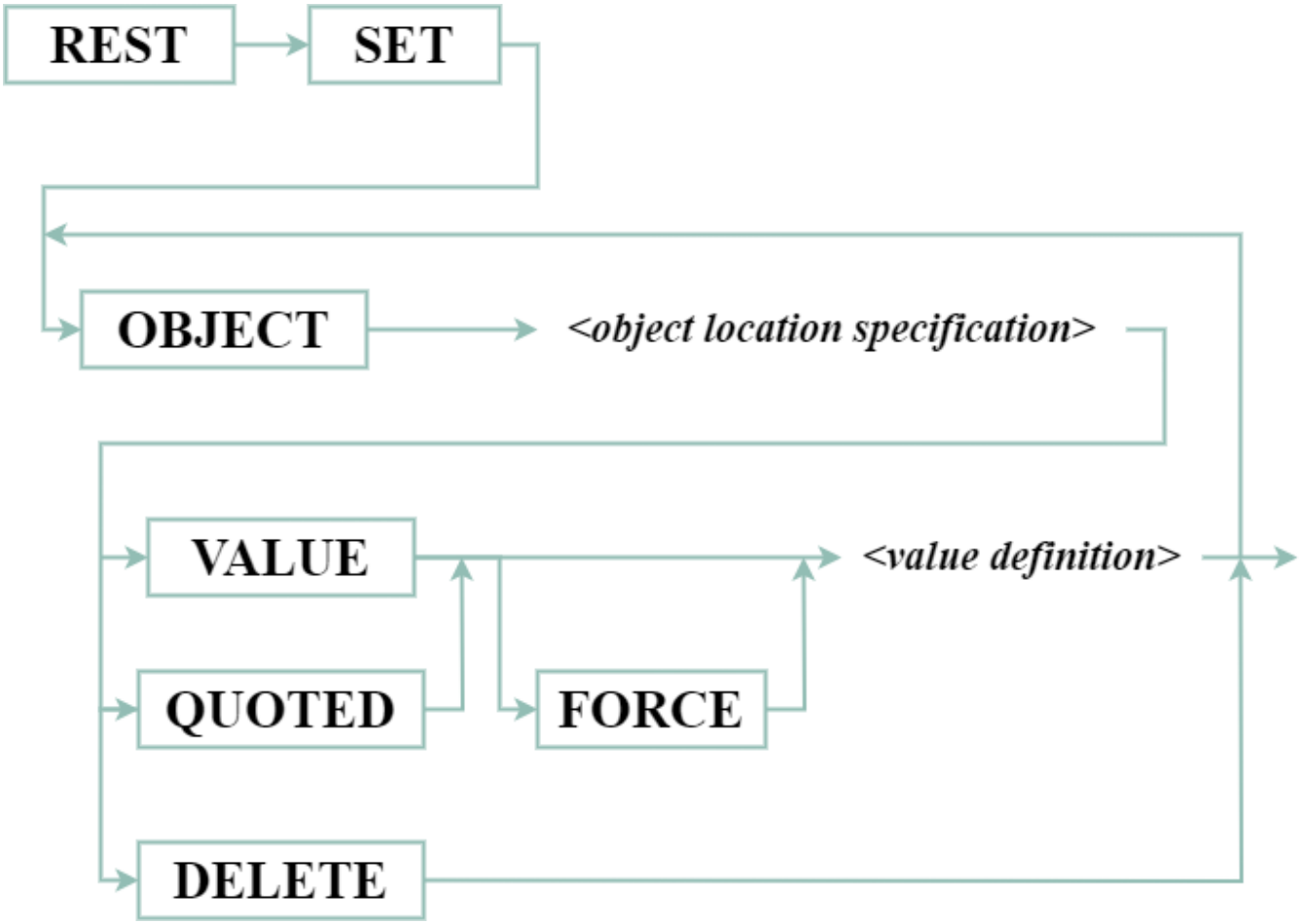
XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the object's value through this translation table from right to left. Chapter Using conversion tables gives more information about tables in general.
OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, interprets the object's value as a floating point number and replaces it with the sum of the value and the offset.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, interprets the object's value as a floating point number and replaces it with the product of the value and the scale factor.
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.

If there are multiple driver variables to be read from a single REST reply sent by the device, you may either use separate `REST PARSE` statements for this or use a single statement with multiple `OBJECT` clauses. Example

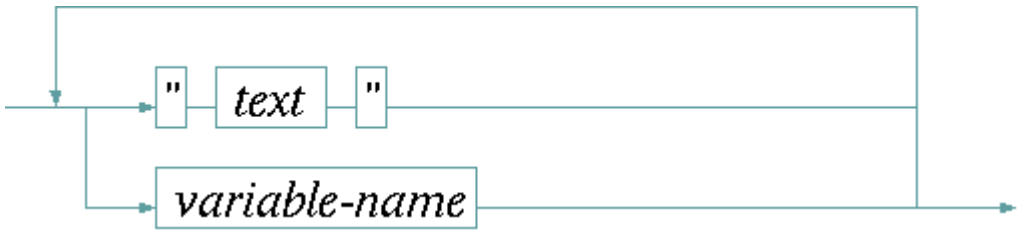
```
REST TRANSACT "GET /api/v1/modulators"
REST PARSE
  OBJECT "/settings/modulator[slot=" config.slotNo "]/frequency"
    DEFAULT "1500.000" VALUE tx.frequency
  OBJECT "/settings/modulator[slot=" config.slotNo "]/fec"
    DEFAULT "3_4" VALUE XLT tFec tx.fec
```

7.2.6.3 The REST SET statement

The `REST SET` statement modified objects in the actual JSON document tree or adds new key / value pairs to the document. The statement starts with the REST SET keywords followed by one or more `OBJECT ... VALUE or OBJECT ... QUOTED` clauses:

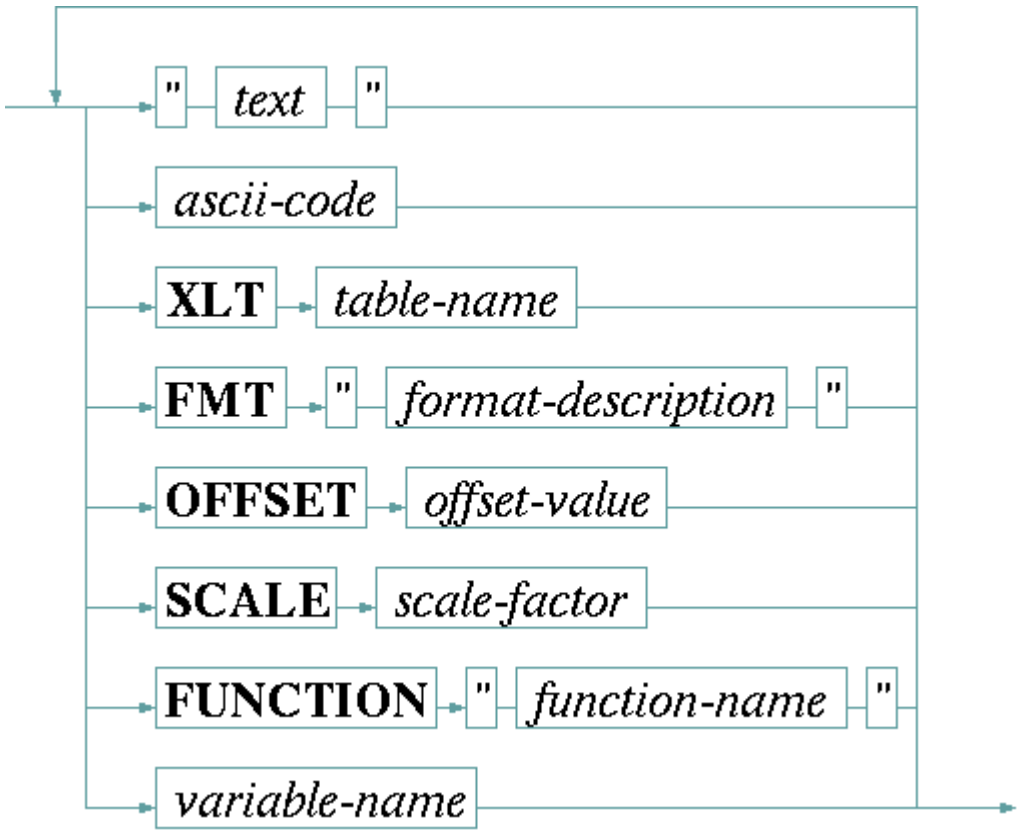


Each object definition starts with the `OBJECT` keyword, followed by the object location definition. The object location definition may be concatenated from multiple quoted strings and ***sat-nms*** driver variables. Section *object paths* in chapter REST I/O functions describes object location definitions more detailed.



Within the REST SET statement, each object is newly created with all its parent objects if it does not yet exist in the document. This includes *conditional* definitions for array elements. If an object already exists in the document, its value will be overwritten by the value specified later in this OBJECT clause.

The object location definition is terminated by the VALUE keyword which starts the definition of the value which shall be assigned to this object. Alternatively, QUOTED may be used instead of VALUE, this is used for textual values which shall be enclosed in double quotes in the JSON text.



The value definition itself is much like writing a command using the PRINT command:

"text"	Plain text, enclosed in double quotes, is added to the output buffer as it is.
ascii-code	A decimal number is interpreted as the ASCII code of a single character to output. You may use this to send special characters like carriage-return or line-feed to the device.

XLT table	The `XLT` keyword, followed by the name of an already defined table, tells the driver to translate the next variable value which shall be printed through this translation table. Chapter Using conversion tables gives more information about tables in general.
FMT "..."	The `FMT` keyword, followed by a format description in double quotes, tells the device driver to format the next variable following the format description given. `FMT` is used to format numeric variables into the representation the device expects in it's remote control command.

A format description consists of the following elements:

format character	description
d b x X f	The first letter of the format description defines the general number representation: `d` (decimal), `b` (binary), `x` (hex, lower case), `X` (hex, upper case) or `f` (floating point)
+	If the `+` option is given, the number is preceded by a +/- character even if it is positive. Together with the `0` option below, the sign appears as an additional character at the first column, hence the field width is increased by one on this case.
0	If the `0` option is given, the field is padded up with zeroes instead of spaces.
1 .. 99	Here follows a one or two digit field width. The field width is the total number of characters the formatted number occupies including the padding characters. If there are more digits needed to show a number correctly, the field with is enlarged automatically. Specifying a field width `1` disables the right orientation in a fixed width completely. This specially is useful for floating point numbers where only the number of fraction digits shall be fixed, not the complete field width.
.	For floating point formats the dot separates the precision from the field width.
0 .. 9	The dot is followed by a one digit specification of the number of fraction digits which shall be printed. The dot and fraction digits specification is valid only with the floating point format.

`INTEGER` variables may be printed using a floating point format, and `FLOAT` variables may be printed with a `d` or `x` format likewise. With `FLOAT` variables you should format in any case as internal precision/rounding problems may cause unpredictable results when floating point values are printed unformatted.

OFFSET o	The `OFFSET` keyword, followed by the (floating point) offset value, tells the driver to add the offset value to the next variable before it is printed.
SCALE s	The `SCALE` keyword, followed by the (floating point) scale factor, tells the driver to multiply the next variable with the scale factor before it is printed
FUNCTION "name"	The `FUNCTION` keyword, followed by the name of the function as a quoted string, tells the driver to apply this function to the next variable before it is printed. The function name is extended by appending `.txt` to achieve the file name to read for the function definition. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
variable-name	A variable name tells the driver to print the value stored in this variable. Usually the *commanded* value is used rather than the value which has been read back from the device. If, however, there has been never a value commanded, or if this variable is a read only variable, the value recently read from the device is used.

Before the variable is printed to the output buffer, any `XLT`, `FMT`, `OFFSET` or `SCALE` operations which have been specified are applied. This happens in a fixed order:

1. If a `SCALE` operation has been specified, this is done first.
2. The `SCALE` operation is followed by an `OFFSET` addition.
3. The so scaled value is formatted according to a FMT specification if one is given.
4. Finally, the value gets translated through a translation table, if an `XLT` operation has been specified.

This scheme applies to numeric (`INTEGER`, `HEX`, `FLOAT`) and to text type variables as well. As soon as a `SCALE`, `OFFSET` or `FMT` keyword is present in front of a variable, this gets converted to a floating point number. Strings which cannot be converted give a zero value.

Example

```
REST CLEAR
  REST SET OBJECT "/settings/frequency" VALUE FMT "f1.3" tx.frequency
  REST SET OBJECT "/settings/fec" QUOTED XLT "tFec" tx.fec
TRANSACT "PATCH /api/v1/modulators?slot=2"
```

Value quoting

In JSON, textual values are enclosed in double quotes, numeric and boolean values are not. The ***sat-nms*** Software does this automatically by looking at the value to set if this is a boolean, a number or a text. This may fail, if the device expects a number enclosed in double quotes.

To cope with this, the `REST SET` statement lets you force the usage of quotes in several ways:

1. Enclose the value in quotes when sending it. You may use single quotes instead of double quotes to avoid the need of escaping the double quote with a backslash. The JSON document sent to the device will contain double quotes instead of the single ones used in the driver. With constraint definitions in the object path this is the only way to force quotes if required.
2. For the value set in a JSON object, you may use the `QUOTED` keyword instead if `VALUE`. The value will be forced to be set in double quotes when added to the document. This way produces better readable code than putting the quote characters explicitly in the sequence of operands after `VALUE`.

Conditional execution

By default a `VALUE` or `QUOTED` clause in the `REST SET` statement causes the JSON document only to be modified if

- The value to be set is a constant
- The variable to be set is marked as *to be written* by the driver. This is if the driver was commanded to set this value, but this did not yet happen.

This behavior protects the JSON document content from unwanted changes if e.g. only one parameter shall be changed but the `REST SET` statement contains the `OBJECT` .. `VALUE` pairs of all parameters in this group.

You can force values to be written regardless of the conditions described above by introducing the keyword `FORCE` directly after the `VALUE` or `QUOTED` keywords.

Object deletion

When used with `DELETE`, `REST SET` deletes the JSON object specified in the `OBJECT` clause and all of it's child objects.

Parent objects are removed as well if they become empty after deletion of the specified object. Please note, that array elements may still not be empty - even if all known child elements of the array element have been deleted. In this case the array element is not automatically removed, must be deleted explicitly by another `REST SET ... DELETE` statement.

7.2.6.4 The REST CLEAR statement

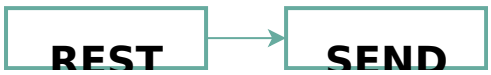
The `REST CLEAR` statement creates an empty JSON document im memory, which deletes the document left by the recent REST TRANSACT statement.

`REST CLEAR` is commonly used to prepare a JSON document from scratch with the `POST` or `PATCH` HTTP methods.

7.2.6.5 The REST SEND statement

The `REST SEND` statement takes the actual JSON document and sends it to the device. It is

used with protocols other than HTTP and replaces `REST TRANSACT` there.

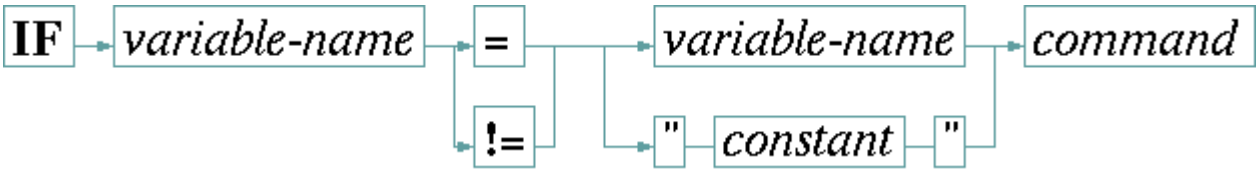


7.2.7 Conditional execution

The device driver language defines a couple of statements to control the flow of execution within a procedure. There is an IF statement as well as a GOTO which refers to a label placed elsewhere in the procedure.

7.2.7.1 The IF statement

The `IF` statement conditionally executes a single statement if the comparison following the `IF` keyword is true.



The keyword `IF` is followed by a comparison of a variable to another variable or constant. With the `=` operator the command is executed if the compared values match, with `!=` if they don't match respectively. Please note, that a constant value must be enclosed in double quotes, even if it is a numeric value. `IF` compares the string representation of the variables after formatting them as defined in the `VAR` statement. Hence, floating point variables always are compared after being rounded to the number of digits defined for the user interface display.

Example

```
IF info.version = "2.0" PRINT "TFR " FMT "f1.3" tx.frequency
IF info.version != "2.0" PRINT "TFR " FMT "d08" SCALE 1000.0 tx.frequency
```

Depending on the value of the `info.version` variable, the example above uses different commands to set the `tx.frequency` at the device. `PRINT` is considered as a single command, with all it's parameters in this case. If more than one command in sequence shall be executed conditionally, a GOTO statement must be used to jump over the this command sequence.

7.2.7.2 The GOTO statement

The `GOTO` statement branches to another location in the same procedure. The destination where to branch to is referenced by a label . The `GOTO` statement lets the execution of the procedure continue at the statement following the label.

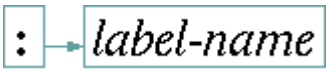
The label, the `GOTO` statement refers to may be defined above or below the location of the `GOTO` statement. However, the label must be defined in the same procedure. The device driver interpreter does not support branches across procedure boundaries.

Some words about loops

Principally `GOTO` statements may be used to create loops within a procedure. You are strongly discouraged from doing so. This is because all procedures for a device (more exactly spoken the procedures of all devices operated at the same physical interface) are executed sequentially. If now one procedure is caught in a loop for some time or perhaps forever, no other procedures will run. The device driver will appear to be *frozen*.

7.2.7.3 Label definition

Labels are defined as destination locations for the GOTO statement. The colon, followed by a label name defines a label:



Labels only are visible within the procedure where they are defined. As a consequence, you may define labels with the same name in different procedures. Within one procedure however, a label name may be used only once. Between the colon and the label name no whitespace is required.

Example

```
IF info.version = "2.0" GOTO v2stuff
   PRINT "TFR " FMT "d08" SCALE 1000.0 tx.frequency
   GOTO finished
:v2stuff
   PRINT "TFR " FMT "f1.3" tx.frequency
:finished
```

7.2.8 Using subroutines

Procedures may be defined to act as subroutines (see chapter The PROC statement for the syntax description). They are not bound to variables in this case. The CALL statement is used to invoke a subroutine from another procedure.

Subroutines are suitable to code operations which have to be done in the same way in several procedures. A common application for this is the check of the device's reply to a command. A subroutine can read the device's reply, check if it is OK and raise a fault flag if not. After a procedure sends a command to the device, it calls the subroutine which does the check.

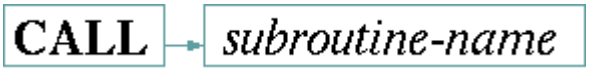
When using subroutines, you should notice the following:

- Subroutines may be nested, this means a subroutine may contain a CALL statement to another subroutine.

- Subroutines must be defined before they can be referenced in another procedure.
- The device driver language allows to define recursive subroutines (subroutines which call themselves). You are strongly discouraged to program recursive subroutines as a program bug in your device driver may cause infinite recursion which definitely will crash the whole MNC application!

7.2.8.1 The CALL statement

The `CALL` statement is used to invoke a subroutine from another procedure. The called routine must be defined before it can be called.



The driver executes all statements of the called procedure and then resumes

Examples

The example below, taken from the Tandberg-Alteia device driver, shows a common application of a subroutine. The subroutine `getAck` is called every time the driver sent a command to the IRD to check if the unit accepted the command.

```
// called after sending a command to the IRD. checks the ACK response which is
// expected.
//
PROC SUBROUTINE getAck
   INPUT "=" TRM "_" internal.ack
   IF internal.ack = "ACK" GOTO endif
      SET faults.commstat = "ACK expected"
      SET faults.99 = "true"
   :endif
```

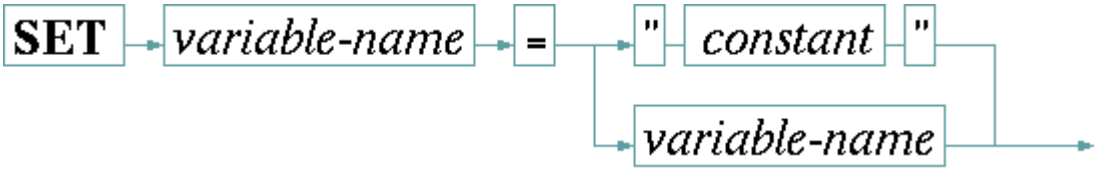
7.2.9 Manipulating variables

The ***sat-nms*** device driver definition language provides a number statements to manipulate driver variables within procedures. These are:

SET	Assigns a value to a variable.
BITSET	Extracts a single bit from a variable and assigns it to another.
RANGESET	Changes the range definition of a variable.
BITSPLIT	Splits up a variable in single bits. Should no longer be used.
BITMERGE	Merges several variables each containing one bit to another variable. Should no longer be used.

7.2.9.1 The SET statement

The `SET` statement lets you assign a value to a variable. The value may be a constant in double quotes (even numeric values must be quoted) or the contents of another variable.



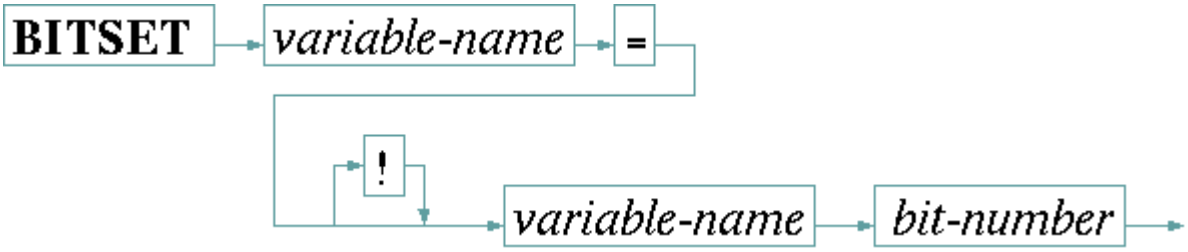
`SET` assigns the value to the internal memory in the variable which remembers the value read from the device. Assigning a value with set hence has the same effect as assigning it within a INPUT or READ statement. `SET` does not change the commanded value a variable receives from the user interface.

Example

```
SET tx.power = "33.2"
```

7.2.9.2 The BITSET statement

Some devices report their fault state as a number in which each bit represents one fault flag. The `BITSET` statement is used to decode flags from such a status value. Usually the fault status is read into an internal variable if the driver. Then the fault bits are decoded from this internal variable using the `BITSET` statement.



The bit position zero addresses the least significant bit in the source variable. The destination variable gets set to `1` or `true` if the addressed bit is set, to `0` or `false` if not. If the variable name is preceded by an exclamation mark, this mapping is inverted.

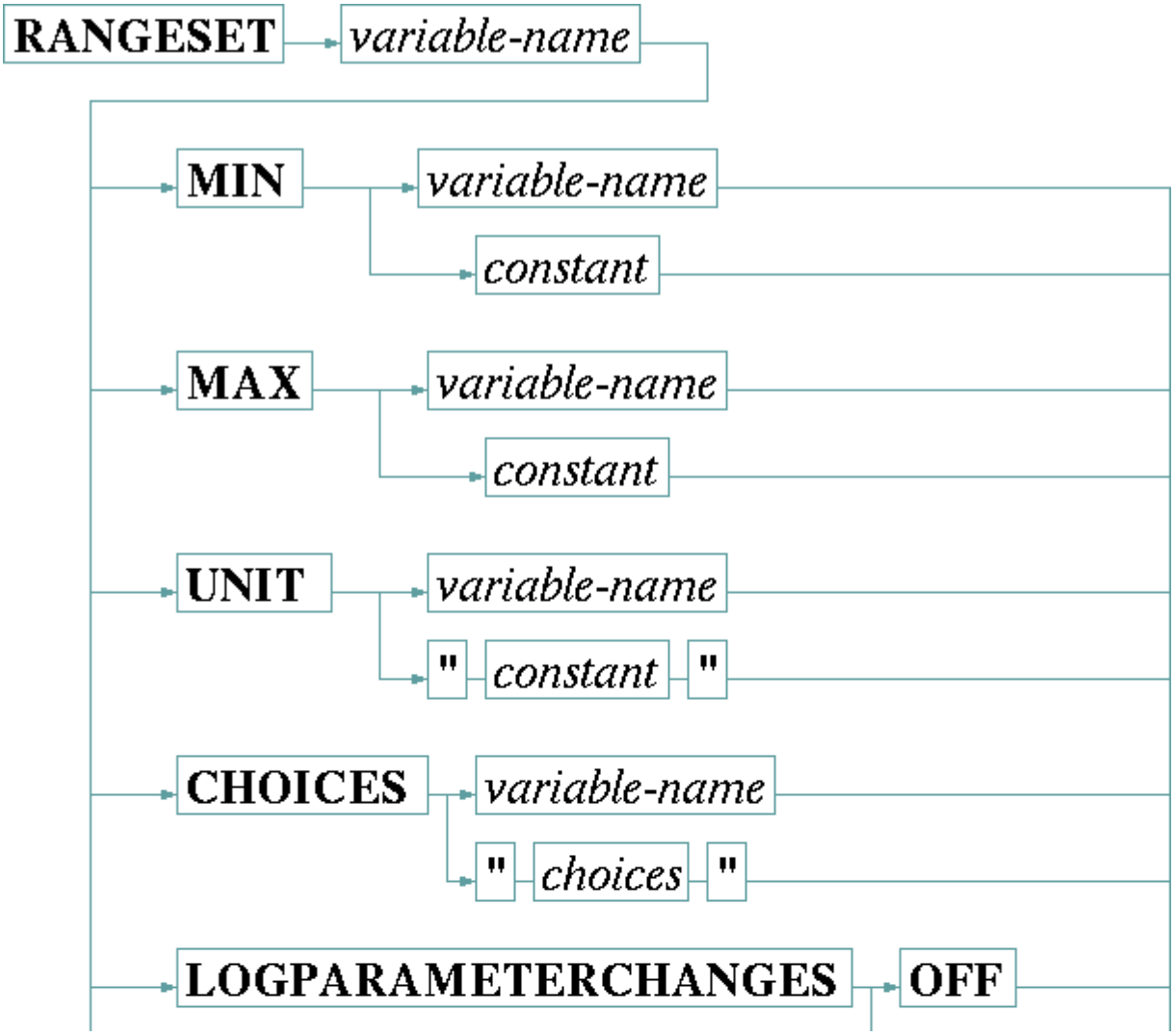
Example

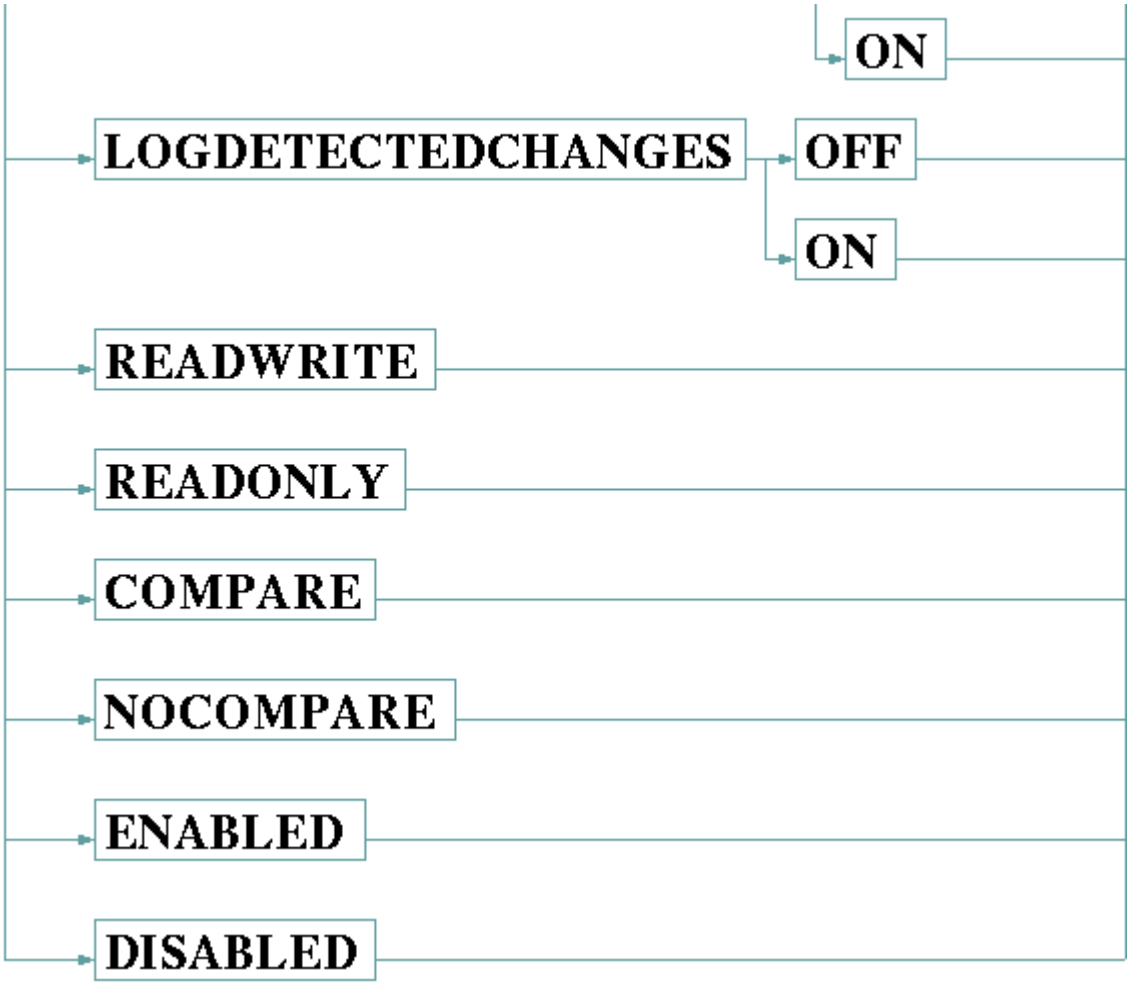
```
INPUT internal.flags
BITSET faults.01 = internal.flags  4    // Temperature
BITSET faults.02 = internal.flags  5    // Signal level
BITSET faults.03 = ! internal.flags  11 // Video lock (inv)
BITSET faults.04 = ! internal.flags  9  // Audio lock (inv)
BITSET faults.05 = internal.flags  13   // High BER
BITSET faults.06 = internal.flags  15   // Demodulator lock
BITSET faults.07 = internal.flags  14   // Conditional Access
```

The example above reads a value into the variable `internal.flags`. BITSET statements are used to extract seven faults flags from this variable. `internal.flags` must be declared `READONLY` to make this example work.

7.2.9.3 The RANGESET statement

The `RANGESET` statement changes several aspects of the range definition of a variable. It is intended to be used in drivers which read some capabilities (e.g. a frequency range) from the device itself.





The options which may follow the `RANGESET` statement are described in the table below:

MIN	Changes the minimum value for a numeric (`INTEGER` , `HEX` , `FLOAT`) variable. The new minimum value may be a constant numeric value or the contents of another variable. `MIN` applies to numeric variables only.
MAX	Changes the maximum value for a numeric (`INTEGER` , `HEX` , `FLOAT`) variable. The new maximum value may be a constant numeric value or the contents of another variable. `MAX` applies to numeric variables only.

UNIT	Changes the maximum value for a numeric (`INTEGER` , `HEX` , `FLOAT`) or a `ALARM` flag variable. The new unit string may e a constant value enclosed in double quotes or the contents of another variable. `RANGESET` / `UNIT` applied to `ALARM` flags changes the alarm description text. To make the unit string for a numeric variable look like the unit specified in the VAR statement, start the string with a space character (e.g. `dBm`). The `VAR` statement implicitly prepends this space character.
CHOICES	Changes the set of choices for a `CHOICE` variable. May be followed either by a variable name or by a quoted string defining the new list of choices. The list of choices is a comma separated list of strings, either defined as a constant enclosed in double quotes or read from the contents of the variable referenced after the `CHOICES` keyword. `CHOICES` applies to `CHOICE` variables only.
READWRITE	Makes the parameter writable from the user interface, overrides a former `READONLY` definition. `READWRITE` applies to all types of variables.
READONLY	Makes the parameter read only. `READONLY` applies to all types of variables.
LOGPARAMETERCHANGESON/OFF	Defines if this variables shall log changes which are commanded to it. Parameter changes are logged if this is set ON and the `logParameterChanges` switch in the device setup page is activated as well. `LOGPARAMETERCHANGES` is `ON` by default for all variables, hence you only need to state `LOGPARAMETERCHANGES OFF` for variables you want explicitly to be excluded from logging.

LOGDETECTEDCHANGESON/OFF	Defines if this variables shall log changes which are detected when reading back the values. Detected changes are logged if this is set `ON` and the `logDetectedChanges` switch in the device setup page is activated as well. `LOGDETECTEDCHANGES` is `ON` by default for all variables, hence you only need to state `LOGDETECTEDCHANGES OFF` for variables you want explicitly to be excluded from logging. Please note, the this modifier has no effect on read-only variables, they are not logged regardless of the logDetected setting.
COMPARE	Enables the read after write comparison check for this variable. This check generates a log message, if the value of the variable read back from the device differs from the commanded value. By default the read after write comparison check is enabled for all variables unless they are defined with the `NOCOMPARE` option in the VAR statement.
NOCOMPARE	Disables the read after write comparison check for this variable.
ENABLED	Enables a formerly disabled variable. `ENABLED` applies to all types of variables.
DISABLED	Disables a variable. Disabled variables show no value at the user interface, any user input is blocked. Disabled variables never cause a procedure which watches this variable to run. `DISABLED` applies to all types of variables.

Updating the range of a variable marks the variable to be read from the device as soon as possible.

The ***sat-nms*** user interface recognizes range definitions every time a window is opened. An already opened window does not change the range definitions of it's input elements. A driver should change variable ranges only due to information which is read once on power up or after a device has been switched on.

Example

```
PRINT "RMAF" INPUT "=" internal.rx.fmax
PRINT "RMF"  INPUT "=" internal.rx.fmin

RANGESET rx.frequency MAX internal.rx.fmax
RANGESET rx.frequency MIN internal.rx.fmin
```

The example above - taken from the SSE K-Star device driver - reads the valid frequency range for the device into the internal driver variables `internal.rx.fmax` and `internal.rx.fmin`. `RANGESET` statements then update the range definition for the `rx.frequency`. This is done in a procedure which is called once after the driver gains communication to the device.

7.2.9.4 The BITSPLIT statement

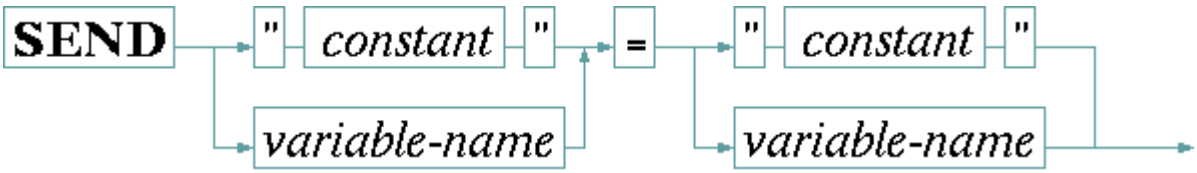
The `BITSPLIT` statement splits up a numeric variable in single bits. `BITSPLIT` will no longer be supported by future versions of the software and therefore should not be used in new device drivers.

7.2.9.5 The BITMERGE statement

The `BITMERGE` statement sets a numeric variable from a set of other variables each controlling one bit in the destination variable. `BITMERGE` will no longer be supported by future versions of the software and therefore should not be used in new device drivers.

7.2.9.6 The SEND statement

The `SEND` statement builds a parameter message from an address and value part and sends this to another device for execution, just like if the parameter had been issued by an operator of the software. The main purpose of this statement is to build logical devices interacting between other device by means of the universal driver language.



The syntax of the `SEND` statement looks much like the SET statement: Following the `SEND` keyword, you specify the destination and the value to be set, separated by `=` character. Both, the destination definition and the value to be sent, may either be quoted string or local variables of this driver:

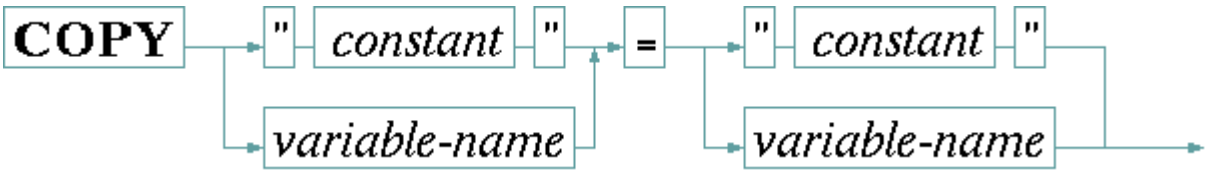
Quoted String	Variable

Quoted String	Variable
Destination	The message destination is literally specified like `DEVICE1.parameterName`. The contents of the variable (e.g. `config.destinationId`) is used as the destination ID for the parameter message. The variable must contain a string value representing a valid message ID
Value	The value to be set at the destination parameter is literally specified. Please note, that also numeric values must be enclosed in quotes here to be recognized al a literal constant.The contents of the variable is send as the value of the parameter message.

At execution time, the `SEND` statement checks if the destination parameter exists and converts the type of the value to be set as required. If this fails, you will find a `SEND to xxxx failed` informational message in the event log in this case.

7.2.9.7 The COPY statement

The `COPY` statement copies either a single parameter or the complete setting from one device to another



The syntax of the `COPY` statement looks much like the `SET` statement: Following the `COPY` keyword, you specify the destination and the source to be copied, separated by `=` character. Both, the destination definition and the value to be sent, may either be quoted string or local variables of this driver.

Quotes strings specify the device names or parameter IDs literally, when specifying variable names, the contents of the variable at execution time is used. Please note that it not possible to copy a complete device setup to a singel variable and vice versa. If the source is specified as a plain device name, the destination must be plain device name as well.

When copying single parameters, a new message is sent to the destination ID with the actual value of the source parameter. No type checking or conversion is done, the value is copied as it is.

When copying a complete device setup, all settings of the source device are collected to an unnamed / temporary device preset. This preset is applied to the destination device. Hence, all parameters the source device stores in presets are copied to the destination.

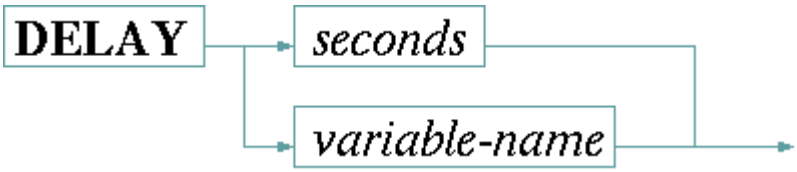
7.2.10 More statements

The ***sat-nms*** device driver definition language contains some more statements which have not yet been described in this document. They are:

DELAY	Pauses the device driver execution for some time.
LOG	Adds a message to the event log.
DRATE	Computes an interface data rate from a symbol rate.
SRATE	Computes a symbol rate from a interface data rate.
WRITEHEX	Outputs a variable containing a hex dump string as binary data.
READHEX	Reads binary data from the device and formats this as a hex dump string into a variable.
INVALIDATE	Invalidates a variable, marks it to be read back as soon as possible.
SYNC	Syncs a variable, updates the value to be commanded with the value recently read back from the device.
RAISECOMMFAULT	Makes the driver to raise a communication fault.

7.2.10.1 The DELAY statement

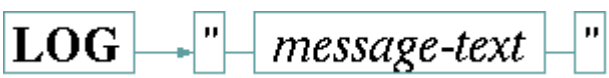
The `DELAY` statement pauses the driver execution for a given number of seconds. The delay may be specified as a (floating point) constant or a variable may be referenced which contains the time to delay.



You should use `DELAY` statements with care. Delaying the driver execution not only slows down the execution of the current procedure, it delays the polling cycle of all devices operated at this serial interface. With longer delay times, the MNC system may no longer be able to recognize device faults within a reasonable time.

7.2.10.2 The LOG statement

The `LOG` statement lets you add an arbitrary message to the MNC system's event log. This may be used to signal events which shall not be treated as a fault, but are to be recorded anyhow.



Messages generated by the `LOG` statement by default are of the priority `INFO`. If the first

character of the message to send is `2` or `3`, the first character is removed from the message and the priority id increased to `FAULT` (2), `ALARM` (3) or `WARNING` (4) respectively.

Example

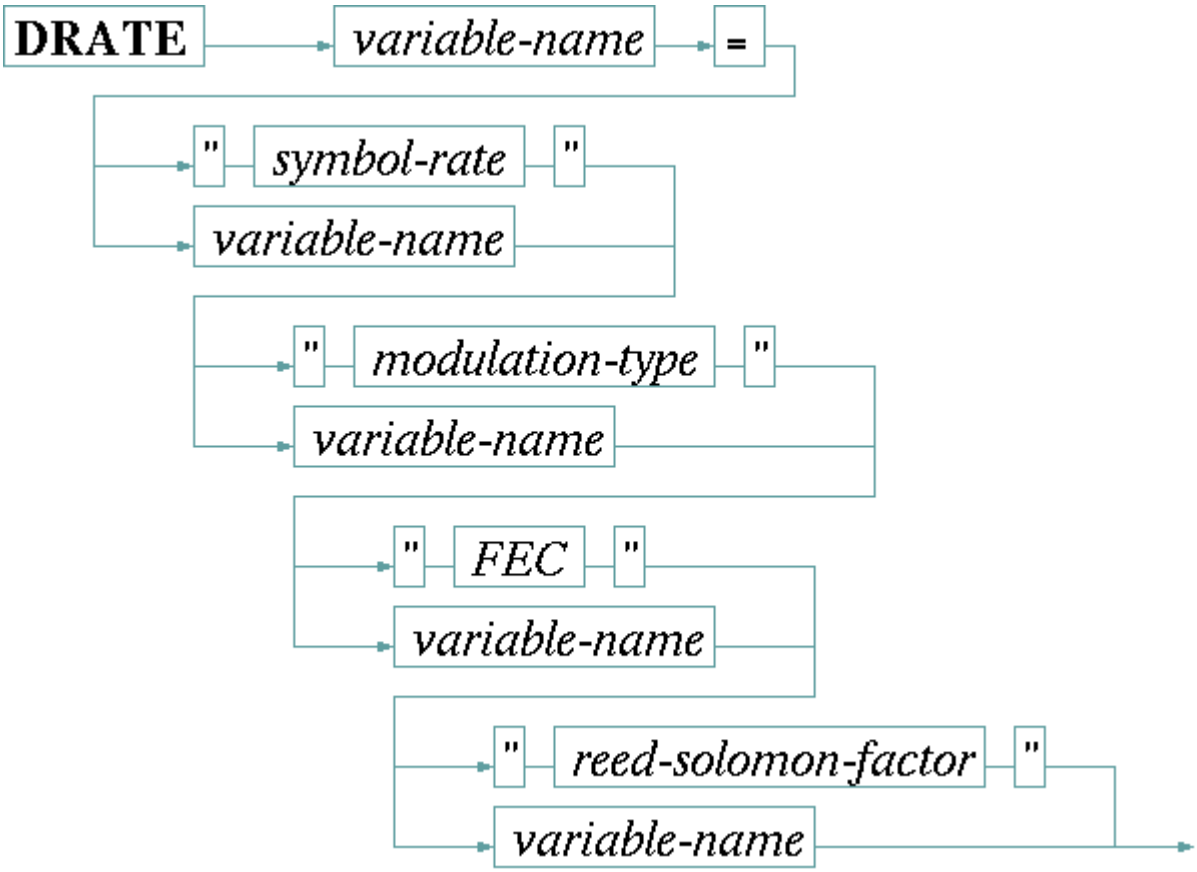
```
LOG "This is an informational message"
LOG "3This is an ALARM message !"
```

The example above adds two messages to the log. The message `This is an informational message` is added with priority `INFO`, the message `This is an ALARM message!` is added with `ALARM` priority.

Please note: although the priority `WARNING` is represented by the number 4, it's logical priority is between `INFORMATIONAL` and `FAULT`. This is because warnings have been added to the software lately and there had do be defined an unused number for the new priority.

7.2.10.3 The DRATE statement

The `DRATE` statement implements a data rate calculator which lets you convert a symbol rate to a interface data rate. This may be useful if a device driver shall supply both values, but the device itself only supports one of these settings.



The `DRATE` statement collects a symbol rate value, a modulation type, a FEC value and a

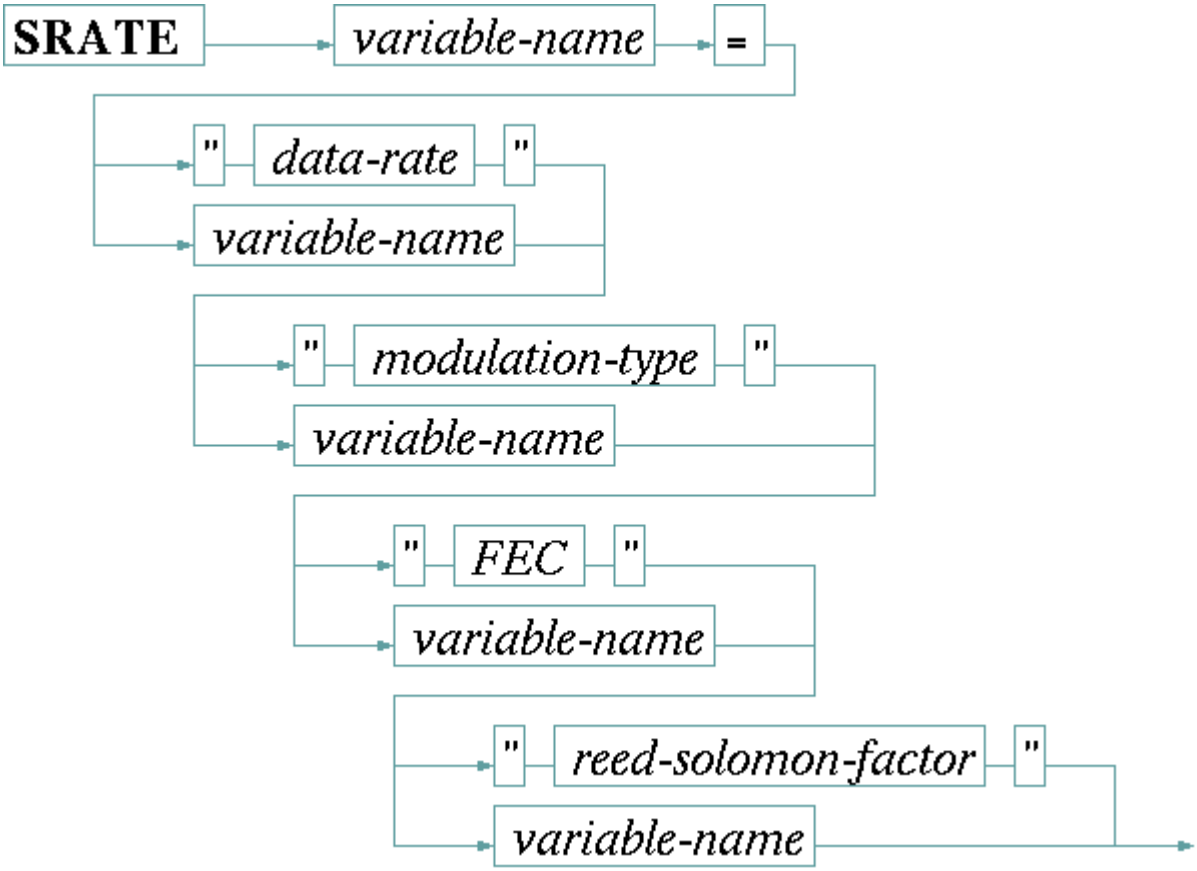
Reed-Solomon factor, computes an interface data rate from this and stores the result into the destination variable. The input data for the DRATE statement may be from constants given in double quotes or from the contents of driver variables.

The `DRATE` statement tries to guess the conversion factors from the input data following the rules described below:

- **modulation type**
  - Tries to guess the modulation type from the characters the token starts with. This catches all common names for modulation types.
  - `16*` : factor 4
  - `8*` : factor 3
  - `Q*` : factor 2
  - Everything else factor 1
- **FEC**
  - May be one of `1/2` , `2/3` , `3/4` , `4/5` , `5/6` , `6/7` , `7/8` or `8/9` .
  - Everything else is treated as `1/1` (no FEC).
- **Reed-Solomon factor**
  - Accepts any factor written as `nnn/mmm` , e.g. `188/204` .
  - If a single number is given, a denominator is assumed to be 204
  - Everything else is treated as `1/1` (no Reed-Solomon)

7.2.10.4 The SRATE statement

The `SRATE` statement implements a data rate calculator which lets you convert a interface data rate into a symbol rate. This may be useful if a device driver shall supply both values, but the device itself only supports one of these settings.



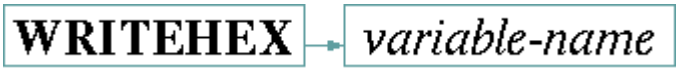
The `SRATE` statement collects a data rate value, a modulation type, a FEC value and a Reed-Solomon factor, computes the symbol rate from this and stores the result into the destination variable. The input data for the `SRATE` statement may be from constants given in double quotes or from the contents of driver variables.

The `SRATE` statement tries to guess the conversion factors from the input data following the rules described below:

- **modulation type**
  - Tries to guess the modulation type from the characters the token starts with. This catches all common names for modulation types.
  - `16* : factor 4
  - `8*` : factor 3
  - `Q*` : factor 2
  - Everything else : factor 1
- **FEC**
  - May be one of `1/2` , `2/3` , `3/4` , `4/5` , `5/6` , `6/7` , `7/8` or `8/9` .
  - Everything else is treated as `1/1` (no FEC).
- **Reed-Solomon factor**
  - Accepts any factor written as `nnn/mmm` , e.g. `188/204` .
  - If a single number is given, a denominator is assumed to be 204.
  - Everything else is treated as `1/1` (no Reed-Solomon)

7.2.10.5 The WRITEHEX statement

The WRITEHEX statement interprets the contents of a variable as a hex dump of a binary byte array and sends this binary data to the device after packing it into a protocol frame.



The WRITEHEX statement is used in the StandardBin.nc include file to define a convenient method of sending a command through the lowLevel driver variable for binary communication protocols. There is probably no other use for the WRITEHEX statement.

7.2.10.6 The READHEX statement

The READHEX statement reads a message from the device, strips off the protocol frame and formats the received binary data as a hex dump. This is assigned to a driver variable.

The READHEX statement is used in the StandardBin.nc include file to define a convenient method of reading back the device's reply through the lowLevel driver variable for binary communication protocols. There is probably no other use for the READHEX statement.

7.2.10.7 The INVALIDATE statement

The INVALIDATE statement marks a variable to be read from the device as soon as possible. It can be used if for instance a general mode setting has been changed at the device - requiring all parameter depending on this mode to be updated. INVALIDATE bypasses the *delayed read back* mechanism controlled by the config.readBackDelay setting, forces the read back in the same cycle if the GET procedure for the variable is located after the INVALIDATE statement within the driver.

For the INVALIDATE statement, use the keyword INVALIDATE followed by the name of the variable to be invalidated.

7.2.10.8 The SYNC statement

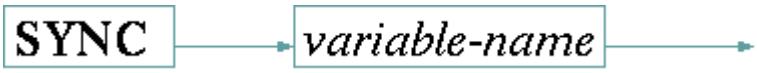
The SYNC statement *syncs* a variable, meaning that the variable's commanded value gets updated with the value recently read back from the device.

Normally, the device driver remembers a variable's value as it has been commanded from the GUI or from a logical device. When called in a PRINT or WRITE statement, this value will be used in favour of the value read back from the device. The driver uses this to check, if the commanded values has been accepted by the device.

When the device alters a commandable parameter autonomously, it may be desirable to accept

this value as the new value to be commanded. For example, imagine an antenna tracking controller device which accepts the pointing angles only as a triple az/el/pol. The operator commands all three angles to find the satellite and switches on tracking. The antenna controller tracks the satellite and as a consequence alters the pointing angles. If now one of the angles will be set in the GUI by the operator, the other two angles will revert to the values which had been set before tracking was switches on.

Using the SYNC statement after reading back the pointing values will help to avoid this situation.



For the `SYNC` statement, use the keyword `SYNC` followed by the name of the variable to be synced.

7.2.10.9 The RAISECOMMFAULT statement

The `RAISECOMMFAULT` statement causes a communication fault to be raised. This e.g. can be used, if the device returns some value which is not expected or that indicates that the device did not recognize the last command. The fault gets cleared once the driver successfully passed one more cycle.

For the `RAISECOMMFAULT` statement, use the keyword `RAISECOMMFAULT` followed by a quoted string containing a message to be shown with the fault.

7.3 The RPN language extension

The ***sat-nms*** device driver definition language has been designed to be easy to understand and to provide a set of commands which is optimized to program device drivers. It enables people with some technical understanding but without any programming experience to add new device drivers to the ***sat-nms*** system.

Some complex devices require the device driver to behave more intelligent than the device driver definition language can do. For such cases the RPN language extension has been added to the device driver language.

The RPN language extension adds some functionality of an RPN (RPN = Reverse Polish Notation) programmable pocket calculator to the device driver. With RPN commands a device driver procedure can do:

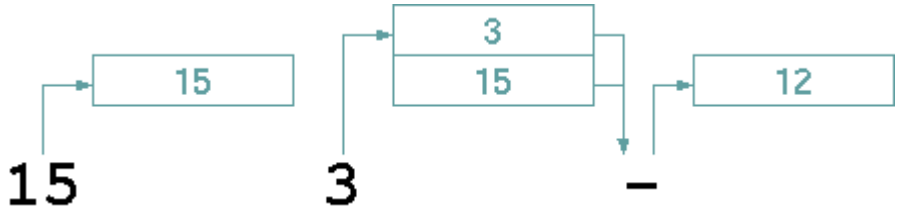
- Arithmetic operations.
- String manipulation operations.
- Text based I/O to the device.
- Loops, conditional branches.

Device drivers using RPN commands are not as easy to understand as standard device drivers are. Only experienced programmers should use these commands as bad designed RPN command sequences may compromise the stability of the whole MNC system.

7.3.1 The RPN stack

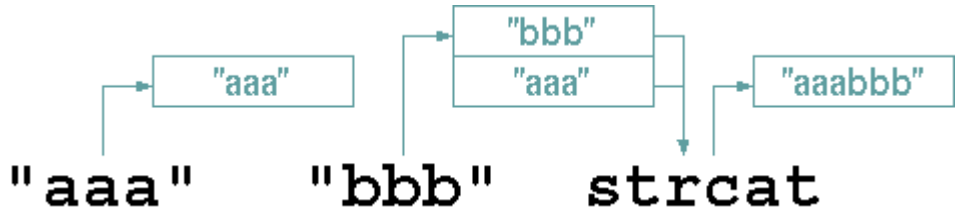
RPN (Reverse Polish Notation) is a way to define arithmetic operations which very efficiently can be used for computer programs. A number of pocket calculators use this method as well as programming languages like FORTH.

The example below illustrates how RPN works. Subtracting 3 from 15 you will be used to write as `15 - 3`. This is known as *infix* notation, the operator (`-`) is written in between the arguments it works on. With RPN (aka. *postfix* notation) the operator appears behind it's arguments:



- Writing `15` pushes this number on the stack.
- Now `3` pushes the second argument on the stack, the number `3` now appears on top of the `15`.
- Finally the `-` operator removes both numbers from the stack and replaces them by the result of the operation.

With the ***sat-nms*** device driver language, RPN operations are not limited to plain arithmetic. The example below illustrates a string concatenation:



The stack is able to store objects of different data types. Numbers, boolean values and character strings may be used. RPN functions automatically convert the type of their input data as required. The stack generally is not limited in size (the computer's memory size however limits the stack).

RPN operations may take an arbitrary number of parameters from the stack and they may leave an arbitrary number of results there. This makes functions possible which are much more complex than a simple `+` or `-`. Furthermore, the stack may be used as a storage for intermediate results. This together allows to perform quite complex operations very efficiently.

There is one stack for each device which keeps it's contents between driver procedures and

cycles. While this feature enables you to do very sophisticated things with the stack, you should be careful not to leave accidentally something on the stack at the end of an operation. The stack may grow with each cycle of the device driver, eating up slowly the whole available memory. Then, after hours, the MNC application aborts due to a lack of memory. There is a `clr` RPN command which clears the stack, you should call it at the end of an RPN sequence.

7.3.2 The { ...} statement

A sequence of words enclosed in curly braces is recognized by the device driver as an RPN command sequence. Within the RPN sequence commands must be separated by whitespace. Each word is interpreted as one of the following:

Variable names	cause the interpreter to push the variable's contents onto the stack.
Table names	cause the interpreter to remember this translation table and use it for the next `rxlt` / `xlt` command(s).
Numeric (decimal) constants	are pushed onto the stack as double precision floating point values.
Character strings (in double quotes)	are pushed as they are.
Commands / keywords	are executed as the RPN command reference in the following chapter describes.

Example

```
{
   "ENQ;AUN" prt                    // request the number of
   inp "=" find                     // audio streams, get the reply
   !internal.numAudio               // and store it

   internal.numAudio if
    ""                              // audio stream list
    0                               // loop counter
    do
     "ENQ;AUX=" over "d04" fmt strcat prt  // request one entry
     swap                           // get the list on top
     inp "=" find                   // the complete entry
     dup "," trm 2 substr           // entry number 2 digits
     " " strcat                     // delimiter
     swap "," find "," find         // language description
     strcat strcat "\n" strcat      // append the reply
     swap                           // loop counter on top
     1 +                            // increment it
     internal.numAudio 1 -          // decrement numAudio,
     !internal.numAudio             // loop until there are more
    internal.numAudio while         // audio streams
    drop                            // the loop counter
   else
    "NONE\n"
   endif
   !audioList
   clr                              // due to paranoia ;-)
}
```

The example above - taken from the Tandberg-Alteia device driver - fills the variable `audioList` with a newline separated list of available audio streams as reported by the IRD. The RPN sequence first gets the number of available streams from the IRD and stores this into `internal.numAudio`, Then, unless `internal.numAudio` is zero, the program builds the list from the information returned by the IRD for each stream.

7.3.3 RPN command reference

The table below summarizes all commands and operations which are accepted by the device driver in RPN mode. The second column of the table describes how a command changes the stack. Left of the `--` the objects the operation takes from the stack are listed. The TOS (top of stack) is the rightmost value. Right of the '--' the results the operation leaves on the stack are shown.

command	stack usage	description

command	stack usage	description
variable-name	**-- v**	Pushes the contents of the variable onto the stack. The type of the variable is retained (`CHOICE` parameters are represented by string values).
numeric-constant	**-- v**	A numeric value (only decimal notation is allowed) is pushed as a floating point number onto the stack.
%numeric-constant	**-- v**	A numeric value (decimal) preceded by a `%` character is pushed as a long integer number onto the stack.
#hex-constant	**-- v**	A hexadecimal numeric value preceded by a `#` character is pushed as a long integer number onto the stack.
"text"	**-- v**	A text in double quotes is pushed onto the stack as a string constant.
! *variable-name*	**v --**	The exclamation mark, followed by a variable name, tells the driver to take one value from the stack an store it into the variable. Type conversion and range check is done as defined at the VAR statement which created the variable.
^ *variable-name*	**v --**	The caret, followed by a variable name works much like store operator described above. However, while the store (`!`) operator refers to the value the device driver read from the device, the caret operator loops the value to the driver as if an operator had commanded it.
<	**x y -- b**	Compares `x` and `y` numerically. The result is `true` if $x < y$.
=	**x y -- b**	Compares the string representation of x and y. The result is `true` if $x = y$. Be careful when comparing floating point values.
>	**x y -- b**	Compares `x` and `y` numerically. The result is `true` if $x > y$.
-	**x y -- z**	Subtracts two values. The result is $x - y$.
/	**x y -- z**	Divides two values. The result is x / y. With $y = 0$ the result always is zero (although this is mathematically not correct).
*	**x y -- z**	Multiplies two values.
+	**x y -- z**	Adds two values.
@	**'s' -- z**	Replaces the name of a parameter by it's value. The parameter name must be a complete parameter ID, the value is that one displayed at the GUI, not the recently commanded one. If `@` fails, the string `0` ist left on the stack.

command	stack usage	description
binand	**x y -- z**	Performs a bitwise AND of two numeric operands.
binneg	**x -- z**	Binary negates the value on the stack.
binor	**x y -- z**	Performs a bitwise OR of two numeric operands.
binxor	**x y -- z**	Performs a bitwise XOR of two numeric operands.
bit	**n p -- b**	Isolates a bit value from a numeric value. Leaves a boolean flag on the stack which is true if the bit number `p` of the value `n` was set. Bit number 0 denotes the least significant bit in `n` .
clr	**... v --**	Clears the stack.
copy	**'s' 'd' --**	Copies a single parameter or a complete device setup from one device to another. `s` , `d` are interpreted as source and destination of the operation, must both be either plain device names or fully qualified parameter IDs (e.g. `MYDEVICE.grp.parName`). When copying complete device setups, this is done via the device preset facility: a temporary / unnamed device preset is taken from the source device an applied to the destination device.
cut	**'s' n -- 's'**	Cuts the string `s` to the length of `n` characters. If `n` is greater than the length of the string to cut, the latter is leaved unchanged.
debug	**x --**	Takes a numeric value from the stack. Turns on debugging output if non-zero. The debugging output can be watched on the debug console window of the MNC software.
do	**--**	Starts a **do ... while** or a **do ... until** loop.
drop	**v --**	Removes one value from the stack.
dup	**v -- v v**	Duplicates the value on top of the stack.
else	**--**	Part of the **if ... else ... endif** construct.
endif	**--**	Closes an **if ... endif** or an **if ... else ... endif** construct.
find	**'s' 'p' -- 's'**	Finds the pattern `p` in the string `s` . Removes the beginning if `s` including the first occurrence of the pattern `p` . The result starts with the character following the first occurrence of `p` in `s` . If `p` is not found in `s` , find leaves an empty string on the stack.

command	stack usage	description
fmt	**n 'fmt' -- 's'**	Formats a numeric value according to format specification given in `fmt`. Leaves the formatted number as a string on the stack. The format specification follows the same rules the FMT option of the PRINT statement does.
func	**v 'fname' -- v**	Processes a numeric value through a tabular function using linear interpolation. First takes the name of the function to apply from the stack, then the value to process. The interpolated value id pushed back to the stack. Works much like the FUNCTION clause available in most I/O statements of the driver language. Chapter Function tables in I/O functions gives an extensive description about functions an how they are used.
hex	**'x' -- n**	Interprets the top of the stack as the string representation of a hexadecimal number. Leaves the numeric value on the stack.
if	**n --**	Start a **if ... endif** or a **if ... else ... endif** clause. Takes one value from the stack. If this value is non-zero, the commands between **if** and **else** are executed.
inp	**-- 's'**	Reads one message from the device and places the data as a string on the stack. The protocol frame is removed.
log10	**x -- z**	This is the inverse function to **pow10**. It takes the common logarithm of the value on the TOS. To use this function to convert a `mW` expressed power to `dBm` follow this example: `mwval log10 10 *` converts the contents of `mwval` from `mW` to `dBm`.
not	**v -- v**	Negates the top of stack. Works with boolean values, numeric values (non-zero is turned into 0, 0 is turned into 1), and common string codings of boolean states.
over	**a b -- a b a**	Copies the value located one below the TOS on top of the stack.
pow	**x y -- z**	Raises `x` to the power of `y`. May be used to compute the square of a number (`3 2 pow` leaves `9` on the stack). Floating point numbers are allowed for both operands, hence you may compute a square root like this: `myvar 0.5 pow` leaves the square root of the contents of myvar on the stack.
pow10	**x -- z**	Raises 10 to the power of `x`. This mainly is intended to convert RF power values expressed in `dBm` to `mW`. Example: `dbmval 10 / pow10` converts the contents of `dbmval` from `dBm` to `mW`, leaves the latter value on the stack.

command	stack usage	description
prt	**'s' --**	Takes a value from the stack and sends its string representation to the device after adding the protocol frame.
rot	**a b c -- b a c**	Rotates three values on top of the stack.
table **rxlt**	**'s' -- 's'**	Translates a string value through the given translation table. Translates from right to left. Uses the table with the name defined somewhere before the **rxlt** command. The table is *not* pushed to the stack, so the exact position of the table name does not matter.
send	**'v' 'd' --**	Sends a message to another device, setting a parameter at this device. `d` is interpreted as the full parameter ID of the message destination (e.g. `MYDEVICE.grp.parName`), `v` is the value to be set there.
shl	**'x' 'y' -- 'z'**	Shifts `x` left by `y` positions. treats all values as long integers.
shr	**'x' 'y' -- 'z'**	Shifts `x` right by `y` positions. treats all values as long integers.
strcat	**'s' 's' -- 's'**	Concats two string values.
substr	**'s' n -- 's'**	Removes `n` character from the beginning of a string value.
swap	**a b -- b a**	Swaps two values on top of the stack.
trm	**'s' 'p' -- 's'**	Cuts the string `s` at that point where the pattern `p` is found in the string. Leaves the string `s` unchanged if `p` does not exist in `s`
until	**n --**	Closes a **do ... until** loop. Takes one value from the stack, loops again if this is zero.
while	**--**	Closes a **do ... while** loop. Takes one value from the stack, loops again if this is non-zero.
table **xlt**	**'s' -- 's'**	Translates a string value through the given translation table. Translates from left to right. Uses the table with the name defined somewhere before the **xlt** command. The table is *not* pushed to the stack, so the exact position of the table name does not matter.

command	stack usage	description

7.4 SNMP device classification and variable binding

As of version 3.1.610 the ***sat-nms*** MNC provides an universal SNMP agent which publishes a limited number of device parameters in a fixed, table oriented MIB. This SNMP agent permits to monitor and partially control any ***sat-nms*** MNC using the same MIB - regardless of the device configuration of the particular MNC.

The MIB provides one table which contains all devices which are configured in the ***sat-nms*** MNC. No special mapping is necessary for this table as there are only very common parameters like the device name, its fault state or its administrative state in this table.

Beside this, the MIB contains a couple of tables, each representing all devices belonging to a so called *device class*. For instance, there is one table containing all HPAs in the MNC and one table containing all waveguide switches. These class specific tables contain some parameters which are common to the devices of the particular class. The HPA table contains parameters for tx-on, for attenuation and forthe measured output power for each device, the waveguide switch table provides a parameter for the switch position.

As device drivers sometimes use different parameter names for the same or similar function, an individual mapping from the *universal* parameter in the MIB to the real parameter in the device is necessary for every single ***sat-nms*** device driver. Device drivers which do not contain any information about the device class this device belongs to and how the parameters are to be mapped into the MIB only appear in the in the general device table of the MIB.

A ***sat-nms*** device (physical or logical) may be represented by multiple device classes in the MIB or by multiple instances of the same class. For example a combined upconverter and HPA may appera as a HPA in this table (with its HPA specific parameters) and a second time in the upconverter table with its frequency parameter.

Also devices consisting of multiple modules can be treated this way in the MIB, there are quad block upconverter on the market which provide four independent converter modules in one device. Such a device can be mapped in way, that it appears as four BUC devices in that table.

7.4.1 Device driver syntax for SNMP definitions

SNMP definitions are coded in the device driver file as special comment lines. This ensures, that new device drivers which include SNMP definitions can be run on older ***sat-nms*** systems which where delivered before this SNMP feature was introduced.

SNMP definitions involve two keywords, `@class` and `@varbind` . `@class` makes the device member of a certain device class and creates a row for tis device in the table of the referenced device class. `@varbind` maps a SNMP OID to a certain ***sat-nms*** driver variable. All `@varbind` definitions following a `@class` definition refer to this and may only map OID names which are valid for the device class defines above.

The example below explains how to define the SNMP mapping for a device:

```
// @class   HPA
// @varbind mncHpaTxOn        tx.on
// @varbind mncHpaAttenuation  tx.attn
// @varbind mncHpaOutputPower  tx.measuredLevel
// @varbind mncHpaHelixCurrent meas.iHelix
// @varbind mncHpaState       state
```

The `@class` definition identifies the device as a power amplifier and creates an instance for the device in the HPAs table. `@class` and the class name must appear in the same line and there must not be any trailing text right of the class name in the line. A list of valid class names may be found in chapter List of SNMP device classes and OID namesThe `@varbind` definitions below map the OIDs defined for one row of the HPA table to the corresponding variables in the driver. Each `@varbind` keyword must be followed by a valid OID name for this device class and then by the name of a device driver variable. Again, all three words must appear in the same line to be recognized correctly.

As mentioned above, a device may be part of multiple device classes at a time, it also may create multiple rows in the same SNMP table to map identical modules in the device to 'copies' of the device in the table. The example below illustrates this for a dual block upconverter:

```
// @class   BUC 1
// @varbind mncBucTxOn       tx.on.1
// @varbind mncBucInputAtten  tx.attn.1
// @class   BUC 2
// @varbind mncBucTxOn       tx.on.2
// @varbind mncBucInputAtten  tx.attn.2
```

Two `@class` definitions create two rows in the BUC table of the MIB. The number 1, 2 following the class name set the module index in the table, this can be used to differentiate between the two BUCs.

Device drivers maintained by SatService always list all `@varbind` OIDs defined for the `@class` of the device. Parameters which have no mapping for the particular device driver are listed with `!varbind` instead of `@varbind` . This prevents the parser from binding this OID to a variable of the driver, at the same time the complete list of `@varbind` OIDs stays visible in the device driver text for reference.

With the mncTarSelTargets `@varbind` (links the target names defined by a SatService ACU-ODM into the MIB), the software uses a `*` wildcard to denote the position of the target index in

the ***sat-nms*** variable name:

```
// @class   TARSEL
// @varbind mncTarSelMoveToTarget      target.goto
// @varbind mncTarSelTargets           target.*
```

7.4.2 List of SNMP device classes and OID names

The following list shows the SNMP device classes known by the software and the valid OID names for each device class. Most OID names are self-explanatory. The MIB file `satnms-mnc-mib.txt` gives more detailled information about each value.

```
ACU                          -- Antenna controller
   mncAcuAzimuthState           r/o
   mncAcuElevationState         r/o
   mncAcuPolarizationState      r/o
   mncAcuAzimuthTarget          r/w
   mncAcuElevationTarget        r/w
   mncAcuPolarizationTarget     r/w
   mncAcuBeaconFrequency        r/w
   mncAcuBeaconLevel            r/o
   mncAcuMonopulseAzError       r/o
   mncAcuMonopulseElError       r/o
   mncAcuTrackingState          r/o
   mncAcuTargetName             r/o

TRX                          -- Tracking receiver
   mncTrxFrequency              r/w
   mncTrxBeaconLevel            r/o
   mncTrxSignalOkState          r/o
   mncTrxBeaconRelativeLevel    r/o

HPA                          -- Power amplifier
   mncHpaTxOn                   r/w
   mncHpaAttenuation            r/w
   mncHpaOutputPower            r/o
   mncHpaHelixCurrent           r/o
   mncHpaState                  r/o
   mncHpaBeam                   r/w
   mncHpaOutputPowerWatt        r/o
   mncHpaPrimaryTemp            r/o
   mncHpaSecondaryTemp          r/o
   mncHpaCurrent                r/o
   mncHpaOperatingHours         r/o

EIRP                         -- EIRP control device
   mncEirpTxOn                  r/w
```

```
    mncEirpCommandedPower        r/w
    mncEirpMeasuredPower         r/o

BUC                             -- Block upconverter
    mncBucTxOn                   r/w
    mncBucFrequency              r/w
    mncBucAttenuation            r/w

UCONV                           -- Upconverter
    mncUconvTxOn                 r/w
    mncUconvFrequency            r/w
    mncUconvAttenuation          r/w

DCONV                           -- Downconverter
    mncDconvRxOn                 r/w
    mncDconvFrequency            r/w
    mncDconvAttenuation          r/w

TLT                             -- Test loop translator
    mncTltTxFrequency            r/w
    mncTltRxFrequency            r/w
    mncTltAttenuation            r/w
    mncTltLoFrequency            r/w

UPC                             -- Uplink power control
    mncUpcEnable                 r/w
    mncUpcInputDelta             r/o
    mncUpcOutputDelta            r/o

DEICE                           -- De-icing controller
    mncDeiceEnable               r/w
    mncDeiceIceCondition         r/o
    mncDeiceRainCondition        r/o
    mncDeiceState                r/o

DEHYD                           -- Waveguide pressurizer
    mncDehydEnable               r/w
    mncDehydPressure             r/o

WEATHER                         -- Weather monitor
    mncWeatherRain               r/o
    mncWeatherWindSpeed          r/o
    mncWeatherHumidity           r/o
    mncWeatherPressure           r/o
    mncWeatherTemperature        r/o
    mncWeatherWindDirection      r/o
```

```
WGS                              - Waveguide switch
   mncWgsPosition                r/w
   mncWgsWrongPosition           r/o
   mncWgsIndifferentPosition     r/o

SWITCH                           -- On/off or 1:n switch
   mncSwitchPosition             r/w
   mncSwitchWrongPosition        r/o
   mncSwitchIndifferentPosition  r/o

PS-11                            -- 1:1 protection switch
   mncPs11Protection             r/w
   mncPs11ActiveChain            r/w
   mncPs11Switched               r/o

PS-1N                            -- 1:N protection switch
   mncPs1nProtection             r/w
   mncPs1nRedundantChain         r/w
   mncPs1nSwitched               r/o

FOL                              -- Fibre optic link
   mncFolCardFaultState          r/o
   mncFolCardType                r/o
   mncFolCardAdminState          r/w
   mncFolCardName                r/w
   mncFolOptPower                r/o
   mncFolRfPower                 r/o
   mncFolRequestedOptPower       r/w
   mncFolRequestedRfPower        r/w
   mncFolAgc                     r/w
   mncFolAttenuation             r/w
   mncFolTemperature             r/o
   mncFolLnbPower                r/w
   mncFolLnbMeasCurrent          r/o
   mncFolLnbMeasVoltage          r/o

PSU                              -- Power supply unit
   mncPsuUnitFaultState          r/o
   mncPsuTemperature             r/o
   mncPsuVoltage1                r/o
   mncPsuVoltage2                r/o
   mncPsuVoltage3                r/o
   mncPsuVoltage4                r/o

TARSEL                           -- Target selection
   mncTarSelMoveToTarget         r/w
   mncTarSelTargets              r/o
```

```
MOD                              -- Modulator control
   mncModInputStreamId             r/o
   mncModInputStatus              r/o
   mncModFrequency                r/w
   mncModLevel                   r/w
   mncModTxOn                    r/w
   mncModSymbolRate                r/w
   mncModStandard                 r/w
   mncModMode                    r/w
   mncModFec                     r/w
   mncModRollOff                  r/w
   mncModPilots                  r/w
```

7.5 Device driver examples

The following pages contain two examples for a device driver, taken from the***sat-nms*** device driver library. While the first driver is a quite simple implementation, the second driver makes use of the RPN language extension and other features.

- NDSatCom-KuBand-Upconverter
- Tandberg-Alteia

7.5.1 NDSatCom-KuBand-Upconverter Example

```
//
//  Device driver for the ND SatCom Ku-band upconverter.
//
//
//  CHANGE RECORD:
//
//  2001-05-05   1.00  initial version.
//  2001-06-23   1.01  the get procedure also depends on the tx.gain and
//                     the tx.frequency parameter.
//  2001-06-27   1.02  info.type set to NDSatCom-Upconverter
//  2001-08-09   1.03  changed info.type to match the file name.
//
COMMENT   "NDSatCom-KuBand-Upconverter 1.03 010809"
PROTOCOL  Miteq-MOD95
INCLUDE   "drivers/Standard.inc"

/** identification variables ****************************************/

VAR info.type       CYCLE 0   TEXT READONLY INIT "NDSatCom-KuBand-Upconverter"
VAR info.port       CYCLE 0   TEXT READONLY
VAR info.frame      CYCLE 0   TEXT READONLY INIT "Upconverter"
VAR info.if         CYCLE 0   TEXT READONLY
```

```
VAR info.if          CYCLE 0   TEXT READ ONLY

/** configuration variables ***************************************************/

/** MNC variables *************************************************************/

VAR tx.on           CHOICE "OFF|ON|"                    CYCLE 2
VAR tx.gain          FLOAT 0 30.0 1 "dB"                 CYCLE 0
VAR tx.frequency     FLOAT 12750.0 14500.0 3 "MHz"          CYCLE 0

/** internal variables ********************************************************/

/** alarm flags ***************************************************************/

ALARM faults.01      TEXT "Remote access"
ALARM faults.02      TEXT "Synthesizer"
ALARM faults.03      TEXT "LO-A lock"
ALARM faults.04      TEXT "LO-B lock"
ALARM faults.05      TEXT "Power supply"
ALARM faults.06      TEXT "IF-LO level"
ALARM faults.07      TEXT "RF-LO level"

/** overall status / parameter fetch routine **********************************/

TABLE T01 "OFF=1,ON=0"
TABLE T02 "70 MHz=0,140MHz=1"
TABLE T03 "0=1,1=0"
PROC GET WATCH tx.on tx.gain tx.frequency
   PRINT "A"
   INPUT AT 2  SCALE 0.001          tx.frequency
        AT 11 SCALE -0.1 OFFSET 30.0 tx.gain
        AT 15 CUT 1 XLT T03         faults.01
        AT 17 CUT 1 XLT T02         info.if
        AT 19 CUT 1 XLT T01         tx.on
        AT 21 CUT 1                 faults.02
        AT 22 CUT 1                 faults.03
        AT 23 CUT 1                 faults.04
        AT 24 CUT 1                 faults.05
        AT 25 CUT 1                 faults.06
        AT 26 CUT 1                 faults.07

/** set the tx.gain ***********************************************************/

PROC PUT WATCH tx.gain
   PRINT "T" SCALE -10.0 OFFSET 300.0 FMT "d03" tx.gain
   INPUT
   DELAY 1.0
```

```
/** set the rf-on state ************************************************/

TABLE T04 "OFF=M,ON=U"
PROC PUT WATCH tx.on
   PRINT XLT T04 tx.on
   INPUT

/** read / set the frequency *******************************************/

PROC PUT WATCH tx.frequency
   PRINT "F" SCALE 1000.0 FMT "d8" tx.frequency
   INPUT
   DELAY 1.0
```

7.5.2 Tandberg-Alteia Example

```
//
// Device driver for the Tandberg Alteia and Alteia Plus DVB receivers.
//
// CHANGE HISTORY:
//
// 1.00 020505 created.
//

COMMENT   "Tandberg-Alteia 1.00 020505"
PROTOCOL  Tandberg-Alteia
INCLUDE   "drivers/Standard.inc"

/** identification variables *******************************************/

VAR info.type            CYCLE 0 TEXT READONLY INIT "Tandberg-Alteia"
VAR info.port            CYCLE 0 TEXT READONLY
VAR info.frame           CYCLE 0 TEXT READONLY INIT "IRD-Alteia"
VAR info.model           CYCLE 0 TEXT READONLY
VAR info.serial          CYCLE 0 TEXT READONLY
VAR info.ca.casid        CYCLE 0 TEXT READONLY
VAR info.ca.codeversion   CYCLE 0 TEXT READONLY
VAR info.ca.bootversion   CYCLE 0 TEXT READONLY
VAR info.ca.modelno       CYCLE 0 TEXT READONLY
VAR info.ca.hardware      CYCLE 0 TEXT READONLY
VAR info.ca.manufacturer  CYCLE 0 TEXT READONLY
VAR info.ca.download      CYCLE 0 TEXT READONLY

/** configuration variables ********************************************/

VAR config.lbandInputs    CYCLE 0 CHOICE "4,2" SETUP SAVE
VAR config.lnbPower       CYCLE 0 CHOICE "OFF,ON,RST" SETUP
```

```
VAR config.inbPower       CYCLE 0 CHOICE "OFF,ON.,BST" SETUP
VAR config.loFreq         CYCLE 0 FLOAT 0 0 1 "MHz" SETUP
VAR config.berThreshold   CYCLE 0 TEXT SETUP
VAR config.sigThreshold   CYCLE 0 INTEGER 0 255 ""  SETUP
VAR config.errFrame       CYCLE 0 CHOICE "FREEZE,BLACK" SETUP

/** internal variables ****************************************************/

VAR internal.numServices  CYCLE 0 INTEGER 0 0 ""  READONLY
VAR internal.numAudio     CYCLE 0 INTEGER 0 0 ""  READONLY
VAR internal.ack          CYCLE 0 TEXT            READONLY
VAR internal.flags        CYCLE 0 HEX     0 0 ""  READONLY
VAR internal.scnt         CYCLE 0 INTEGER 0 0 ""  READONLY INIT "0"


/** MNC variables *********************************************************/


VAR input              CYCLE 300 CHOICE "1,2,3,4"
VAR polarization         CYCLE 0   CHOICE "HOR,VER"
VAR frequency            CYCLE 0   FLOAT 0 0 1 "MHz"
VAR dataRate             CYCLE 0   FLOAT 0 0 3 "Mbps" READONLY
VAR symbolRate           CYCLE 300 FLOAT 0 0 3 "Msps"
VAR modulation           CYCLE 0   CHOICE "BPSK,QPSK,8PSK"
VAR fec                CYCLE 0   CHOICE "1/2,2/3,3/4,4/5,5/6,6/7,7/8,8/9"
VAR programNo            CYCLE 0   INTEGER 0 99 ""

VAR programList          CYCLE 2   TEXT READONLY
VAR audioList          CYCLE 3   TEXT READONLY
VAR actualProgram        CYCLE 2   TEXT READONLY

VAR video.fmt625         CYCLE 300 CHOICE "PALI,PALB,PALN"
VAR video.fmt525         CYCLE 0   CHOICE "NTSC,NTSN,PALM"
VAR video.level          CYCLE 0   INTEGER -30 30 "%"
VAR video.test           CYCLE 0   CHOICE "NRM,625.1,625.2,625.3,625.4"
                                  "625.5,625.6,625.7,625.8,625.9"
                                  "625.10,625.11,625.12,625.13"
                                  "625.14,625.15,625.16,525.1"
                                  "525.2,525.3,525.4,525.5,525.6"
                                  "525.7,525.8,525.9,525.10,525.11"
                                  "525.12,525.13,525.14,525.15,525.16"

VAR audio.1.program       CYCLE 0   INTEGER 0 0 ""
VAR audio.1.routing       CYCLE 0   CHOICE "NRM,MON,LFT,RGT"
VAR audio.1.output        CYCLE 0   CHOICE "ANALOG,AES/EBU,SPDIF,AC3"
VAR audio.1.level         CYCLE 0   INTEGER 6 18 "dB"
VAR audio.1.language      CYCLE 0   TEXT
VAR audio.1.test          CYCLE 0   CHOICE "NRM,TEST-1,TEST-2,TEST-3,TEST-4,TEST-5"
```

```
VAR audio.1.test        CYCLE 0   CHOICE "NRM,TEST-1,TEST-2,TEST-3,TEST-4,TEST-5
VAR audio.1.info        CYCLE 4   TEXT READONLY

VAR audio.2.program     CYCLE 0   INTEGER 0 0 ""
VAR audio.2.routing     CYCLE 0   CHOICE "NRM,MON,LFT,RGT"
VAR audio.2.output      CYCLE 0   CHOICE "ANALOG,AES/EBU,SPDIF,AC3"
VAR audio.2.level       CYCLE 0   INTEGER 6 18 "dB"
VAR audio.2.language    CYCLE 0   TEXT
VAR audio.2.test        CYCLE 0   CHOICE "NRM,TEST-1,TEST-2,TEST-3,TEST-4,TEST-5"
VAR audio.2.info        CYCLE 4   TEXT READONLY

VAR tsout.routing       CYCLE 300 CHOICE "PRE-CA,POST-CA"
VAR tsout.fibre         CYCLE 0   CHOICE "ON,OFF"

VAR flags.lock          CYCLE 1   BOOL READONLY
VAR flags.ca            CYCLE 0   BOOL READONLY
VAR flags.video         CYCLE 0   BOOL READONLY
VAR flags.audio         CYCLE 0   BOOL READONLY
VAR flags.ber           CYCLE 0   BOOL READONLY

VAR state.ber           CYCLE 0   TEXT          READONLY
VAR state.signal        CYCLE 0   INTEGER 0 255 "" READONLY
VAR state.aspect        CYCLE 3   TEXT          READONLY
VAR state.lines         CYCLE 3   TEXT          READONLY
VAR state.state         CYCLE 2   TEXT          READONLY

VAR ca.status           CYCLE 4   TEXT READONLY
VAR ca.service          CYCLE 0   TEXT READONLY
VAR ca.rasmode          CYCLE 0   CHOICE "DISABLED,FIXED,DSNG,SEC-CA"
VAR ca.bissmode         CYCLE 0   CHOICE "DISABLED,MODE 1,MODE E"
VAR ca.bisskey1         CYCLE 0   TEXT
VAR ca.bisskey2         CYCLE 0   TEXT
VAR ca.bissbits         CYCLE 0   TEXT
VAR ca.dsngkey          CYCLE 0   TEXT

VAR reset               CYCLE 0   TEXT

/** alarm flags ***************************************************************/

ALARM faults.01 TEXT "Temperature"
ALARM faults.02 TEXT "Signal level"
ALARM faults.03 TEXT "Video lock"
ALARM faults.04 TEXT "Audio lock"
ALARM faults.05 TEXT "High BER"
ALARM faults.06 TEXT "Demodulator Lock"
ALARM faults.07 TEXT "Conditional Access"
```

```
/** translation tables ***************************************************/

TABLE tModulation   "BPSK=BPS,QPSK=QPS,8PSK=8PS"
TABLE tErrFrame     "FREEZE=FRZ,BLACK=BLK"
TABLE tVideoLevel   "-30=NEG30,-29=NEG29,-28=NEG28,-27=NEG27,-26=NEG26"
                    "-25=NEG25,-24=NEG24,-23=NEG23,-22=NEG22,-21=NEG21"
                    "-20=NEG20,-19=NEG19,-18=NEG18,-17=NEG17,-16=NEG16"
                    "-15=NEG15,-14=NEG14,-13=NEG13,-12=NEG12,-11=NEG11"
                    "-10=NEG10,-9=NEG09,-8=NEG08,-7=NEG07,-6=NEG06"
                    "-5=NEG05,-4=NEG04,-3=NEG03,-2=NEG02,-1=NEG01"
                    "0=POS00,1=POS01,2=POS02,3=POS03,4=POS04"
                    "5=POS05,6=POS06,7=POS07,8=POS08,9=POS09"
                    "10=POS10,11=POS11,12=POS12,13=POS13,14=POS14"
                    "15=POS15,16=POS16,17=POS17,18=POS18,19=POS19"
                    "20=POS20,21=POS21,22=POS22,23=POS23,24=POS24"
                    "25=POS25,26=POS26,27=POS27,28=POS28,29=POS29,30=POS30"
TABLE tVideoTest    "NRM=00,625.1=01,625.2=02,625.3=03,625.4=04,625.5=05"
                    "625.6=06,625.7=07,625.8=08,625.9=09,625.10=10,625.11=11"
                    "625.12=12,625.13=13,625.14=14,625.15=15,625.16=16"
                    "525.1=17,525.2=18,525.3=19,525.4=20,525.5=21,525.6=22"
                    "525.7=23,525.8=24,525.9=25,525.10=26,525.11=27,525.12=28"
                    "525.13=29,525.14=30,525.15=31,525.16=32"
TABLE tTsRouting    "PRE-CA=PRE,POST-CA=PST"
TABLE tTsFibre      "ON=ENA,OFF=DIS"
TABLE tAudioRouting "NRM=ST,MON=M1,LFT=M2,RGT=M3"
TABLE tAudioOutput  "ANALOG=ANA,AES/EBU=PRO,SPDIF=SPD,AC3=AC3"
TABLE tAudioTest    "NRM=0,TEST-1=1,TEST-2=2,TEST-3=3,TEST-4=4,TEST-5=5"
TABLE tRasMode      "DISABLED=DIS,FIXED=FIX,DSNG=DSN,SEC-CA=SCA"
TABLE tBissMode     "DISABLED=DIS,MODE 1=MO1,MODE E=MOE"
TABLE tCaStatus     "UNKNOWN CODE=FF"
                    "CARD INSERTED=00"
                    "CARD REMOVED=01"
                    "CARD INVALID=04"
                    "SERVICE BLOCKED=05"
                    "INVALID PACKET=06"
                    "CARD UNAUTHORIZED=07"
                    "HARDWARE FAILURE=08"
                    "CLEAR BUT RESTRICTED=09"
                    "SERVICE BLACKED OUT=10"
                    "SERVICE EXPIRED=11"
                    "CA WARNING=12"
                    "CA WARNING=13"
                    "PAIRING ERROR=14"
                    "CA WARNING=15"
                    "CA WARNING=16"
                    "CA WARNING=17"
                    "CA WARNING=21"
```

```
                "CA WARNING=22"
TABLE tCaService    "UNKNOWN CODE=XXX"
                "NO SERVICE SELECTED=NSS"
                "CLEAR=CLR"
                "RAS AUTHORIZED=RAS"
                "RAS UNAUTHORIZED=RAU"
                "BISS AUTHORIZED=BIA"
                "BISS UNAUTHORIZED=BIU"
                "VGUARD AUTHORIZED=GVA"
                "VGUARD UNAUTHORIZED=VGU"
                "NO CA INSTALLED=NCA"

/** procedures  ************************************************************/


// called after sending a command to the IRD. checks the ACK response which is
// expected.
//
PROC SUBROUTINE getAck
    INPUT "=" TRM "_" internal.ack
    IF internal.ack = "ACK" GOTO endif
        SET faults.commstat = "ACK expected"
        SET faults.99 = "true"
    :endif


// called if the number of l-band inputs gets changed. sets the range for the
// "input" parameter
//
PROC PUT WATCH config.lbandInputs
    IF config.lbandInputs = "2" GOTO else
        RANGESET input CHOICES "1,2,3,4"
        GOTO endif
    :else
        RANGESET input CHOICES "1,2"
    :endif

// this procedure is executed once after the communication to the receiver ahs
// been established. it get the device identification parameters and sets the
// LNB frequency according to the position of the input switch.
//
PROC GET WATCH info.serial info.model
    PRINT "REM;MOD"
    INPUT "MOD=" TRM "_" info.model
    PRINT "REM;SNM"
    INPUT "SNM=" TRM "_" info.serial
```

```
// reads the fault flags and derived the "LED flags" from these values
//
//
PROC GET WATCH flags.lock
    PRINT "OPR;SRQ"
    INPUT
      "SIG="          state.signal
      "ALR=" TRM "_"   internal.flags
      "BER=" TRM "_"   state.ber
    BITSET faults.01 = internal.flags  4  // Temperature
    BITSET faults.02 = internal.flags  5  // Signal level
    BITSET faults.03 = internal.flags  11 // Video lock
    BITSET faults.04 = internal.flags  9  // Audio lock
    BITSET faults.05 = internal.flags  13 // High BER
    BITSET faults.06 = internal.flags  15 // Demodulator lock
    BITSET faults.07 = internal.flags  14 // Conditional Access
    IF faults.06 = "true" SET flags.lock = "false"
    IF faults.07 = "true" SET flags.ca = "false"
    IF faults.03 = "true" SET flags.video = "false"
    IF faults.04 = "true" SET flags.audio = "false"
    IF faults.05 = "true" SET flags.ber = "false"
    IF faults.06 = "false" SET flags.lock = "true"
    IF faults.07 = "false" SET flags.ca = "true"
    IF faults.03 = "false" SET flags.video = "true"
    IF faults.04 = "false" SET flags.audio = "true"
    IF faults.05 = "false" SET flags.ber = "true"


// reads the service status message. this also is used to select a service
// after the IRD has decoded the signal
//
PROC GET WATCH state.state
{
    "SER;SRQ" prt inp
    "SRQ=" find ! state.state

    state.state "WAIT.PMT" = if
      internal.scnt 1 + !internal.scnt
      internal.scnt 6 > if
        "SER;SEL=00" prt inp
      else
        internal.scnt 3 > if
          "SER;SEL=" programNo "d02" fmt strcat prt inp
        endif
      endif
    else
      0 !internal.scnt
```

```
    endif
}

// reads the information from the TUN:SRQ request.
//
PROC GET WATCH
    input frequency modulation polarization
    config.lnbPower config.loFreq
    PRINT "TUN;SRQ"
    INPUT
        "FRQ=" SCALE 0.125          frequency
        "LNB=" SCALE 0.125          config.loFreq
        "PWR=" CUT 3               config.lnbPower
        "POL=" CUT 3               polarization
        "RFI=" CUT 1              input
        "MOD=" CUT 3 XLT tModulation  modulation
    DRATE dataRate = symbolRate modulation fec "188"

// sets the RF input and the frequency. we switch both together to ensure that
// the input is set before the frequency is tuned.
//
PROC PUT WATCH input frequency
    PRINT "TUN;RFI=NO " input
    CALL getAck
    DELAY 3.0
    PRINT "TUN;FRQ=" SCALE 8.0 FMT "d06" frequency
    CALL getAck

// sets LO frequency (of the input actually set)
//
PROC PUT WATCH config.loFreq
    PRINT "TUN;LNB=" SCALE 8.0 FMT "d06" config.loFreq
    CALL getAck

// sets the modulation type
//
PROC PUT WATCH modulation
    PRINT "TUN;MOD=" XLT tModulation modulation
    CALL getAck

// sets the polarization by a 22kHz pulse
//
PROC PUT WATCH polarization
    PRINT "TUN;POL=" polarization
    CALL getAck

// reads the information from the DEM;SRQ request.
//
```

```
//
PROC GET WATCH fec symbolRate config.berThreshold config.sigThreshold
    PRINT "DEM;SRQ"
    INPUT
        "FEC=" CUT 3              fec
        "SYM=" TRM "_" SCALE 0.0001  symbolRate
        "BER=" TRM "_"             config.berThreshold
        "SIG=" CUT 3              config.sigThreshold
    DRATE dataRate = symbolRate modulation fec "188"

// sets the symbol rate
//
PROC PUT WATCH symbolRate
    PRINT "DEM;SYM=" SCALE 10000.0 FMT "d06" symbolRate
    CALL getAck

// sets the FEC
//
PROC PUT WATCH fec
    PRINT "DEM;FEC=" fec
    CALL getAck

// sets the BER threshold configuration parameter
//
PROC PUT WATCH config.berThreshold
    PRINT "DEM;BER=" config.berThreshold
    CALL getAck

// sets the signal level configuration parameter
//
PROC PUT WATCH config.sigThreshold
    PRINT "DEM;SIG=" FMT "d03" config.sigThreshold
    CALL getAck

// sets the LNB power configuration parameter
//
PROC PUT WATCH config.lnbPower
    PRINT "TUN;PWR=" config.lnbPower
    CALL getAck

//////////////////// VIDEO PARAMETERS ////////////////////////

// gets the video state (and the video settings)
//
PROC GET WATCH video.fmt625 video.fmt525 video.level video.test
             state.aspect state.lines config.errFrame
    PRINT "VID;SRQ"
    INPUT
```

```
    INPUT
        "625=" TRM "_" video.fmt625
        "525=" TRM "_" video.fmt525
        "ERR=" TRM "_" XLT tErrFrame config.errFrame
        "LVL=" TRM "_" XLT tVideoLevel video.level
        "OPS=" TRM "_" XLT tVideoTest video.test
        "VLS=" TRM "_" state.lines
        "SAR=" TRM "_" state.aspect

// sets the 625 lines default video mode
//
PROC PUT WATCH video.fmt625
    PRINT "VID;625=" video.fmt625
    CALL getAck

// sets the 525 lines default video mode
//
PROC PUT WATCH video.fmt525
    PRINT "VID;525=" video.fmt525
    CALL getAck

// sets what to show is video is missing
//
PROC PUT WATCH config.errFrame
    PRINT "VID;ERR=" XLT tErrFrame config.errFrame
    CALL getAck

// sets the video level
//
PROC PUT WATCH video.level
    PRINT "VID;LVL=" XLT tVideoLevel video.level
    CALL getAck

// sets the video test mode
//
PROC PUT WATCH video.test
    PRINT "VID;OPS=" XLT tVideoTest video.test
    CALL getAck

////////////////////// AUDIO CHANNEL 1 ////////////////////////////

// gets the audio channel 1 parameters
//
PROC GET WATCH audio.1.routing audio.1.output audio.1.level audio.1.language
    audio.1.test audio.1.info
    PRINT "AUD;SRQ"
    INPUT
        "ROU=" CUT 2 XLT tAudioRouting audio.1.routing
```

```
        "ROU="  CUT 2 XLT tAudioRouting audio.1.routing
        "LEV=M" CUT 2 audio.1.level
        "OUT="  CUT 3 XLT tAudioOutput audio.1.output
        "DFL="  CUT 3 audio.1.language
        "OPS="  CUT 1 XLT tAudioTest audio.1.test
        "CLN="  audio.1.info

// set the audio channel 1 output routing
//
PROC PUT WATCH audio.1.routing
    PRINT "AUD;ROU=" XLT tAudioRouting audio.1.routing
    CALL getAck

// set the audio channel 1 level
//
PROC PUT WATCH audio.1.level
    PRINT "AUD;LEV=M" FMT "d02" audio.1.level
    CALL getAck

// set the audio channel 1 output (hardware)
//
PROC PUT WATCH audio.1.output
    PRINT "AUD;OUT=" XLT tAudioOutput audio.1.output
    CALL getAck

// set the audio channel 1 default language
//
PROC PUT WATCH audio.1.language
    PRINT "AUD;DFL=" audio.1.language
    CALL getAck

// set the audio channel 1 test mode
//
PROC PUT WATCH audio.1.test
    PRINT "AUD;OPS=" XLT tAudioTest audio.1.test
    CALL getAck


////////////////////// AUDIO CHANNEL 2 ////////////////////////////

// gets the audio channel 2 parameters
//
PROC GET WATCH audio.2.routing audio.2.output audio.2.level audio.2.language
    audio.2.test audio.2.info
    PRINT "AU2;SRQ"
    INPUT
        "ROU="  CUT 2 XLT tAudioRouting audio.2.routing
        "LEV=M" CUT 2 audio.2.level
```

```
        "OUT="  CUT 3 XLT tAudioOutput audio.2.output
        "DFL="  CUT 3 audio.2.language
        "OPS="  CUT 1 XLT tAudioTest audio.2.test
        "CLN="  audio.2.info

// set the audio channel 2 output routing
//
PROC PUT WATCH audio.2.routing
    PRINT "AU2;ROU=" XLT tAudioRouting audio.2.routing
    CALL getAck

// set the audio channel 2 level
//
PROC PUT WATCH audio.2.level
    PRINT "AU2;LEV=M" FMT "d02" audio.2.level
    CALL getAck

// set the audio channel 2 output (hardware)
//
PROC PUT WATCH audio.2.output
    PRINT "AU2;OUT=" XLT tAudioOutput audio.2.output
    CALL getAck

// set the audio channel 2 default language
//
PROC PUT WATCH audio.2.language
    PRINT "AU2;DFL=" audio.2.language
    CALL getAck

// set the audio channel 2 test mode
//
PROC PUT WATCH audio.2.test
    PRINT "AU2;OPS=" XLT tAudioTest audio.2.test
    CALL getAck


////////////////////// TRANSPORT STREAM OUTPUT ////////////////////////

// reads the transport stream parameters
//
PROC GET WATCH tsout.routing tsout.fibre
    PRINT "TSO;SRQ"
    INPUT
        "CAM=" CUT 3 XLT tTsRouting tsout.routing
        "ENA=" CUT 3 XLT tTsFibre   tsout.fibre

// sets the transport stream pre/post CA routing
```

```
//
PROC PUT WATCH tsout.routing
   PRINT "TSO;CAM=" XLT tTsRouting tsout.routing
   CALL getAck

// enables / disable the ASI fibre output
//
PROC PUT WATCH tsout.fibre
   PRINT "TSO;ENA=" XLT tTsFibre tsout.fibre
   CALL getAck

////////////////////// CONDITIONAL ACCESS ////////////////////////////


// reads the conditional access parameters. uses an RPN command sequence for
// this as - depending on the purchased configuration - the Alteia does not
// report all of these parameters all the time. RPN assigns empty strings to
// parameters which do not exist in the reply.
//
PROC GET WATCH ca.status ca.rasmode ca.bissmode
   ca.dsngkey ca.bisskey1 ca.bisskey2 ca.bissbits
{
   "CAS;SRQ" prt inp
   dup "CSS=" find "_" trm tCaService rxlt ! ca.service
   dup "CST=" find "_" trm tCaStatus rxlt ! ca.status
   dup "CID=" find "_" trm ! info.ca.casid
   dup "CAC=" find "_" trm ! info.ca.codeversion
   dup "BCV=" find "_" trm ! info.ca.bootversion
   dup "CMN=" find "_" trm ! info.ca.modelno
   dup "CHT=" find "_" trm ! info.ca.hardware
   dup "CMA=" find "_" trm ! info.ca.manufacturer
   dup "CDS=" find "_" trm ! info.ca.download
   dup "RAM=" find "_" trm ! tRasMode rxlt ca.rasmode
   dup "BIS=" find "_" trm ! tBissMode rxlt ca.bissmode
   dup "BM1=" find "_" trm ! ca.bisskey1
   dup "BM2=" find "_" trm ! ca.bisskey2
   dup "BKE=" find "_" trm ! ca.bissbits
   dup "DSK=" find "_" trm ! ca.dsngkey
   clr
}

// sets the RAS mode parameter
//
PROC PUT WATCH  ca.rasmode
   PRINT "CAS;RAM=" XLT tRasMode ca.rasmode
   CALL getAck
```

```
// sets the DSNG key used with one of the RAS modes
//
PROC PUT WATCH  ca.dsngkey
    PRINT "CAS;DSK=" ca.dsngkey
    CALL getAck

// sets the BISS mode parameter
//
PROC PUT WATCH  ca.bissmode
    PRINT "CAS;BIS=" XLT tBissMode ca.bissmode
    CALL getAck

// sets the BISS mode 1 key
//
PROC PUT WATCH  ca.bisskey1
    PRINT "CAS;BM1=" ca.bisskey1
    CALL getAck

// sets the BISS mode 2 key
//
PROC PUT WATCH  ca.bisskey2
    PRINT "CAS;BM2=" ca.bisskey2
    CALL getAck

// sets the BISS key length for modes 2/3
//
PROC PUT WATCH  ca.bissbits
    PRINT "CAS;BKE=" ca.bissbits
    CALL getAck

////////////////// SERVICE SELECTION ////////////////////////////////


// reads the name of the actually selected service
//
PROC GET WATCH actualProgram
    PRINT "ENQ;NAM"
    INPUT "NAM=" actualProgram


// reads the service list (the bouquet) from the IRD and distributes the list
// as the variable programList. this proc is completely written in RPN language
// as this supports more flexible building the list from the particular
// entries.
//
PROC GET WATCH programList
{
```

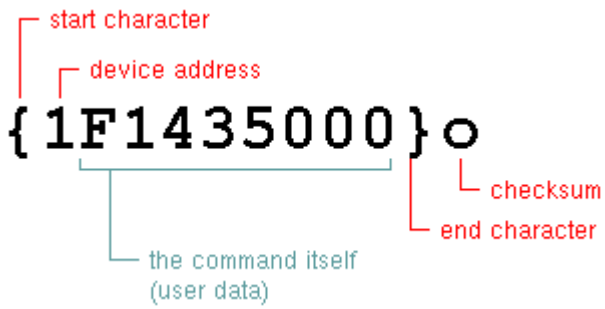
```
    state.state "WAIT.DEM" = if             // in WAIT.DEM state,
      0 !internal.numServices               // no services are available
    else
      "ENQ;NUM" prt                         // request the number of
      inp "=" find                          // services, get the reply
      !internal.numServices                 // and store it
    endif

    internal.numServices if
      "ENQ;NET" prt inp "NET=" find         // network name
      "\n" strcat                           // as the first line of the list
      0                                     // loop counter
      do
        "ENQ;SER=" over "d02" fmt strcat prt  // request one entry
        swap                                // get the list on top
        inp "=" find dup                    // get the reply (2x)
        "," trm                             // the service index
        " " strcat                          // append " " to it
        swap "," find strcat                // append the service name
        strcat                              // append list entry
        "\n" strcat                         // trailing NL
        swap                                // loop counter on top
        1 +                                 // increment it
        internal.numServices 1 -            // decrement numServices,
        !internal.numServices               // loop until there are more
      internal.numServices while            // services
      drop                                  // the loop counter
    else
      "NO PROGRAMS AVAILABLE\n"
    endif
    !programList
    clr                                     // due to paranoia ;-)
}


// selects a service.
//
PROC PUT WATCH programNo
    PRINT "SER;SEL=" FMT "d02" programNo
    CALL getAck


// reads the audio list of the actually selected service from the IRD and
// distributes the list as the variable audioList. this proc is completely
// written in RPN language as this supports more flexible building the list
// from the particular entries.
//
```

```
//
PROC GET WATCH audioList
{
   "ENQ;AUN" prt                      // request the number of
   inp "=" find                      // audio streams, get the reply
   !internal.numAudio                   // and store it

   internal.numAudio if
    ""                              // audio stream list
    0                               // loop counter
    do
     "ENQ;AUX=" over "d04" fmt strcat prt  // request one entry
     swap                             // get the list on top
     inp "=" find                       // the complete entry
     dup "," trm 2 substr                 // entry number 2 digits
     " " strcat                       // delimiter
     swap "," find "," find             // language description
     strcat strcat "\n" strcat           // append the reply
     swap                             // loop counter on top
     1 +                             // increment it
     internal.numAudio 1 -                // decrement numAudio,
     !internal.numAudio                  // loop until there are more
    internal.numAudio while                // audio streams
    drop                             // the loop counter
   else
    "NONE\n"
   endif
   !audioList
   clr                              // due to paranoia ;-)
}

// selects the audio 1 stream
//
PROC PUT WATCH audio.1.program
   PRINT "SER;A1L=" FMT "d05" audio.1.program
   CALL getAck

// selects the audio 2 stream
//
PROC PUT WATCH audio.2.program
   PRINT "SER;A2L=" FMT "d05" audio.2.program
   CALL getAck
```

7.6 Device communication protocols

A device communication protocol encapsulates the recurrent operations to handle things like

start and stop characters or checksums with each message to be sent or received. The example below shows the complete message a ND-SatCom upconverter receives to tune its frequency. The pure command is `F1435000` (sets the frequency to 14,350.00 MHz). The protocol frame starts with a `{` followed by the device address. The command is terminated by `}` and finally a checksum character is sent.



A device communication protocol in the ***sat-nms*** software adds the protocol frame data (here marked red) to each message sent by a device driver with a `PRINT` or `WRITE` statement. The `PRINT` or `WRITE` statement creates the user data, the pure command. The protocol handler looks up the device address in the setup settings for the device, adds the start and stop characters and calculates the checksum over the message. The other way round, for each message received, the protocol handler strips off the protocol frame data and verifies the address and checksum fields. An `INPUT` or `READ` statement then receives the pure command data.

With the ***sat-nms*** software, new protocol definitions may be added to the software, simply by editing a text file for the new type of protocol frame which is needed.

7.6.1 Writing a communication protocol definition

As mentioned above, protocol definitions are coded as simple text files. When the MNC program starts, it *compiles* the protocols needed for it's equipment setup to memory. Writing a protocol definition means editing such a text file and storing it at a place in the MNC computer where the MNC program searches for these files.

In a communication protocol definition file, you compose the protocol frame around a command from elements like characters, checksums or strings using a couple of keywords. The protocol definition language is quite simple, but powerful enough to specify almost all communication protocols used by satcom equipment.

A MNC system keeps all protocol definitions in a subdirectory called `protocols`. On standard installations this is the directory `/home/satnms/protocols`. The names of the protocol files consist of the protocol name followed by the extension `.proto`.

If you are writing the device driver on a Linux based MNC or NMS computer, you may want to use the VI text editor or mcedit (included in midnight commander) text editor for this. Both has been configured to colorize device driver files on these machines. We also providing syntax definitions for Visual Studio Code. Please ask your contact at SatService for details.

You also may copy protocol definition files to a MS-Windows based computer to edit them there. If you do this, you should consider the following:

- Protocol files are Unix based text files. Lines are terminated with a line-feed character only. Your favorite MS-Windows text editor may have problems to show these files.
- Unix / Linux is case sensitive with file names. Be sure that you don't mess up the case of characters in file name when you copy files between Unix and MS-Windows.

7.6.2 General file format

The protocol definition language uses a file format which is very similar to that one used with ***sat-nms*** device drivers. Like with device drivers the following applies:

- Whitespace (space characters, tabs, line breaks) separates words.
- Line breaks have no special termination function.
- All keywords are in upper case letters.
- Comments in C/C++ style are recognized (both, `/* ... */` and `// ...` comments).

Beside this, each protocol definition file has the same simple structure:

- The file starts with an (optional) comment block and some common definitions.
- Then, following the TRANSMIT keyword, the specification how a message shall be composed during transmit is given.
- Then the RECEIVE keyword starts the specification how to parse incoming data.

Below, an example for a protocol definition file is shown.

```
//
//  The communication protocol used with SSE devices in NPI mode.
//

COMMENT  "SSE-NPI 1.00 020108"
CLASS    TTYProtocol


/** packet send procedure ***********************************************/

TRANSMIT

  CHAR       2              // start byte, STX
  CHAR       "F"            // source (master) LU
  CHAR       "F"
  ADDRESS    TEXT           // destination (slave) LU, 2 characters
  HEXLENGTH  0              // 2 characters hex coded data length
  USERDATA                  // the command string
  CHECKSUM   SUM8H 1 -1      // checksum 2 characters
  CHAR       3              // end byte, ETX

/** packet receive procedure ********************************************/

RECEIVE

  START      2              // start byte, STX
  ADDRESS    TEXT           // source (slave) LU
  CHAR       "F"            // destination (master) LU
  CHAR       "F"
  HEXLENGTH  0              // 2 characters hex coded data length
  USERDATA                  // the response string
  CHECKSUM   SUM8H 1 -1 2    // checksum 2 characters
  CHAR       3              // end byte, ETX
```

7.6.3 Global definitions

The first section of the protocol definition file defines the protocol identification string and the Java class which is responsible to perform the I/O operations at runtime.

The `COMMENT` statement defines an identification string which is passed to the user interface. An operator can identify the type and version of a communication protocol used with a certain device.

The text following the `COMMENT` keyword is free field and principally may contain any

information. The device protocols coming with the ***sat-nms*** software all follow a convention which defines the comment string as

"protocol-name X.YY YYMMDD"

where `X` is the major version number of the protocol, `YY` the minor version number and `YYMMDD` the release date of this protocol version. It is recommended that customer defined device protocols follow this scheme, too.

The `CLASS` statement defines the Java class used perform the protocol I/O steps during runtime of the MNC software. Actually `TTYProtocol` is the only protocol class which is applicable for customer defined protocol definitions.

Some communication protocols may be configured in some aspects of their functionality by applying protocol parameters to them. This is done with the `PROTOCOLPARAMETER` statement:

There may multiple `PROTOCOLPARAMETER` statements in one protocol definition file, each defining one aspect of the communication protocol. The `key` defines the parameter to set, `value` is the value to be assigned to this parameter.

Protocol parameters may be set either in the protocol definition file (thus defining the parameter for all uses of the protocol) or in the device driver. A definition in the device driver overwrites the setting in the protocol file.

ATTENTION: If you have multiple devices in the same device thread (`INTERFACE`) using different device drivers, the result is unpredictable if the drivers do not make exactly the same protocol parameter settings.

Example

```
PROTOCOLPARAMETER "post.content.type=application/xml"
```

7.6.4 TX message elements

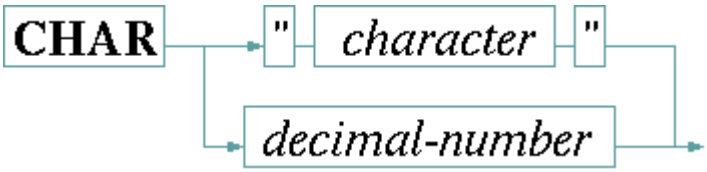
Starting with the `TRANSMIT` keyword, the second section of the protocol definition specifies the steps required to compose a valid protocol frame which is sent to the device.

The protocol frame definition consists of a sequence step specifiers, represented by a keyword followed by a defined number of parameters each. The step specifiers recognized in the the `TRANSMIT` section are:

CHAR	Outputs a single character.
ADDRESS	Outputs the device address.
USERDATA	Outputs the user data.
CHECKSUM	Outputs a message checksum.
DATALENGTH	Outputs a data length field (binary).
HEXLENGTH	Outputs a data length field (hex).
SEQUENCE	Outputs a binary message sequence number.

7.6.4.1 CHAR

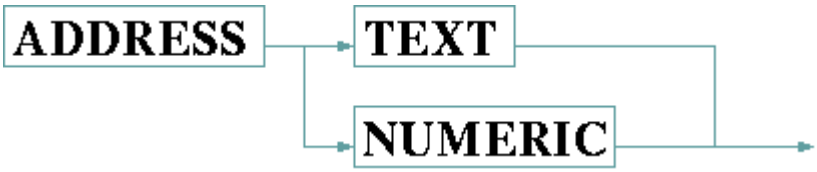
The CHAR step outputs a single character (byte).



The value may be specified as a single character in double quotes or as a decimal number in the range `0 ... 255`.

7.6.4.2 ADDRESS

The `ADDRESS` step outputs the device address.



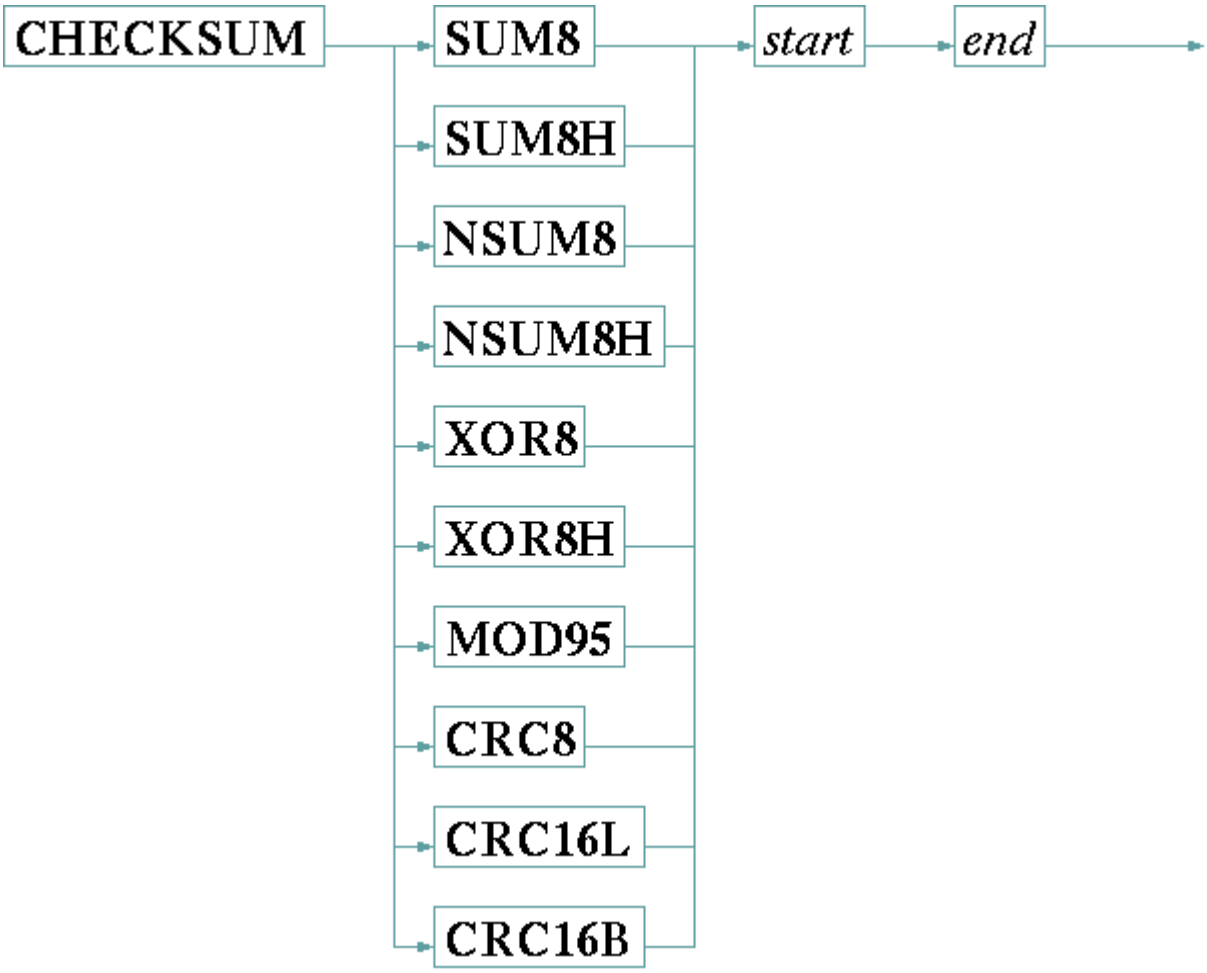
The `ADDRESS` keyword must be followed by one of `TEXT` or `NUMERIC`. If `TEXT` is specified, the address is output as a character string as it is entered at the user interface to the `address` configuration variable. If `NUMERIC` is specified, the address string is converted to a integer number and output as a single byte of this value.

7.6.4.3 USERDATA

The `USERDATA` step copies the user data as it was generated from the `PRINT` or `WRITE` statement in the device driver to the output.

7.6.4.4 CHECKSUM

The `CHECKSUM` step outputs a checksum calculated from the message contents.



The parameters of this step specifier are:

SUM8	Creates a one byte checksum by adding all bytes and truncating the sum to the lowest 8 bits.
SUM8H	Like `SUM8`, but the checksum is coded as a 2 digit hex number.
NSUM8	Creates a one byte checksum by adding all bytes, taking the negative value of the sum and then truncating the sum to the lowest 8 bits.
NSUM8H	Like `NSUM8`, but the checksum is coded as a 2 digit hex number.
XOR8	Creates a one byte checksum by XOR-ing all specified bytes.
XOR8H	Like `XOR8`, but the checksum is coded as a 2 digit hex number.
MOD95	Creates the one character *modulo 95* checksum, known from the protocol used by Miteq devices (see below).

CRC8	Creates a one byte `CRC8` checksum.
CRC16L	Creates a two byte `CRC16` checksum, using the ARC/LHA CRC algorithm. Places the least significant byte first (little endian).
CRC16B	Creates a two byte `CRC16` checksum, using the ARC/LHA CRC algorithm. Places the most significant byte first (big endian).
start	The buffer position of the first character to be included in the checksum computation (0 means the first character).
end	The buffer position of the last character to be included in the checksum computation, relative to the actual position (-1 means the character left of the first checksum character).

The `MOD95` type checksum is computed following the formula:

$$Checksum = 32 + MOD_{95}\left[\left(\sum_{i=1}^{N} message\ byte_i\right) - (32 * N)\right]$$

7.6.4.5 DATALENGTH

The `DATALENGTH` protocol step outputs a packet length field.

The packet length is computed as the length of the user data string created by the `PRINT` or `WRITE` statement in the device driver plus the offset defined as a decimal number. It is coded as a one byte binary.

7.6.4.6 HEXLENGTH

The `LENGTH` protocol step outputs a packet length field.

The length is computed as the length of the user data string created by the `PRINT` or `WRITE` statement in the device driver plus the offset defined as a decimal number. It is coded as a two character hex string.

7.6.4.7 SEQUENCE

The `SEQUENCE` step outputs a one byte binary sequence number.

The sequence number gets incremented with each message that has been sent.

7.6.5 RX message elements

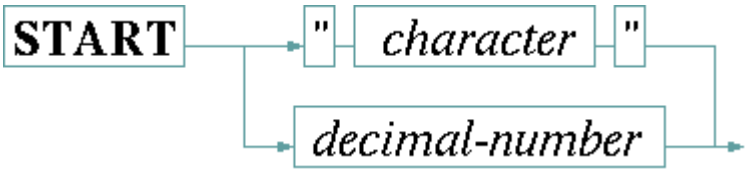
Starting with the `RECEIVE` keyword, the third section of the protocol definition specifies the steps required to parse a protocol frame which is received from the device.

The protocol frame definition consists of a sequence step specifiers, represented by a keyword followed by a defined number of parameters each. The step specifiers recognized in the the RECEIVE section are:

START	Reads a message start character.
CHAR	Reads a single character.
ADDRESS	Reads the device address.
USERDATA	Reads the user data.
STRING	Reads a terminated string as the user data
CHECKSUM	Reads a message checksum.
DATALENGTH	Reads a data length field (binary).
HEXLENGTH	Reads a data length field (hex).

7.6.5.1 START

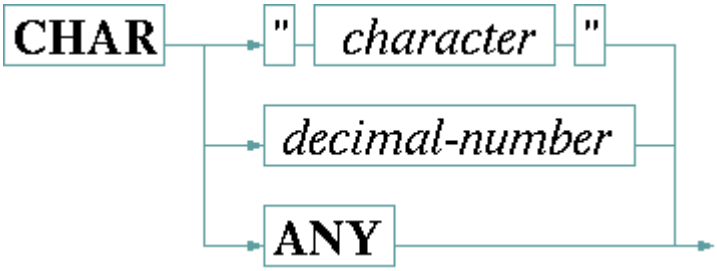
The `START` protocol step reads (and discards) all incoming data until the specified start character is received. The start character is not assumed to be part of the user data.



The character value may be specified as a single character in double quotes or as a decimal number in the range `0 ... 255`.

7.6.5.2 CHAR

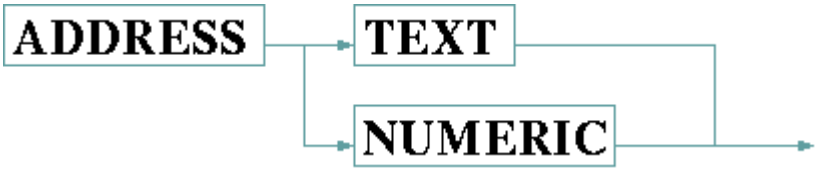
The `CHAR` step reads a single character and compares this to the expected value. If another character than expected has been received, the reception of this message is aborted and a communication fault is reported. The character received is not treated as part of the user data.



The value may be specified as a single character in double quotes or as a decimal number in the range `0 ... 255`. The special value `ANY` may be specified to make the `CHAR` step accept any character.

7.6.5.3 ADDRESS

The `ADDRESS` step reads the device address and compares it to the address value set in the device's setup menu. If the `ADDRESS` step received other data than expected, the reception of this message is aborted and a communication fault is reported.



The `ADDRESS` keyword must be followed by one of `TEXT` or `NUMERIC`:

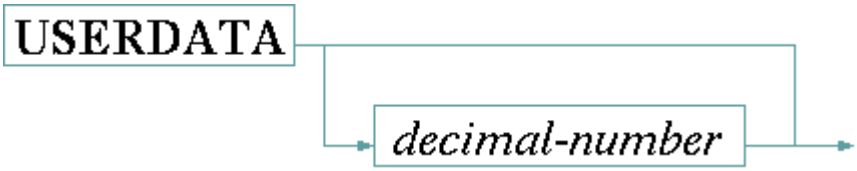
If `TEXT` is specified, the step reads as many characters as the address string configured for this device is long. The address is compared character by character to the received data.

If `NUMERIC` is specified, one character (byte) is read. The address string is converted to a integer number and compared to the received byte.

7.6.5.4 USERDATA

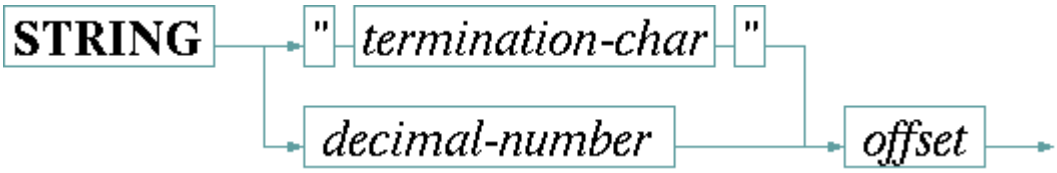
The `USERDATA` step reads a known number of bytes and returns these as the user data of the message. USERDATA requires to know the number of bytes to read, from one of these sources:

- A DATALENGTH protocol step appearing above the `USERDATA` step in the protocol definition.
- A HEXLENGTH protocol step appearing above the `USERDATA` step in the protocol definition.
- A fixed number of bytes, specified by a number following the `USERDATA` keyword. If the protocol definition contains a `DATALENGTH/HEXLENGTH` step and an explicitly given length in the `USERDATA` step as well, the length specified with the `USERDATA` step has precedence.



7.6.5.5 STRING

The `STRING` step reads a character string of variable length which is terminated by a known character. The received data may be truncated by a certain number of character before it is returned as the user data part of the message.



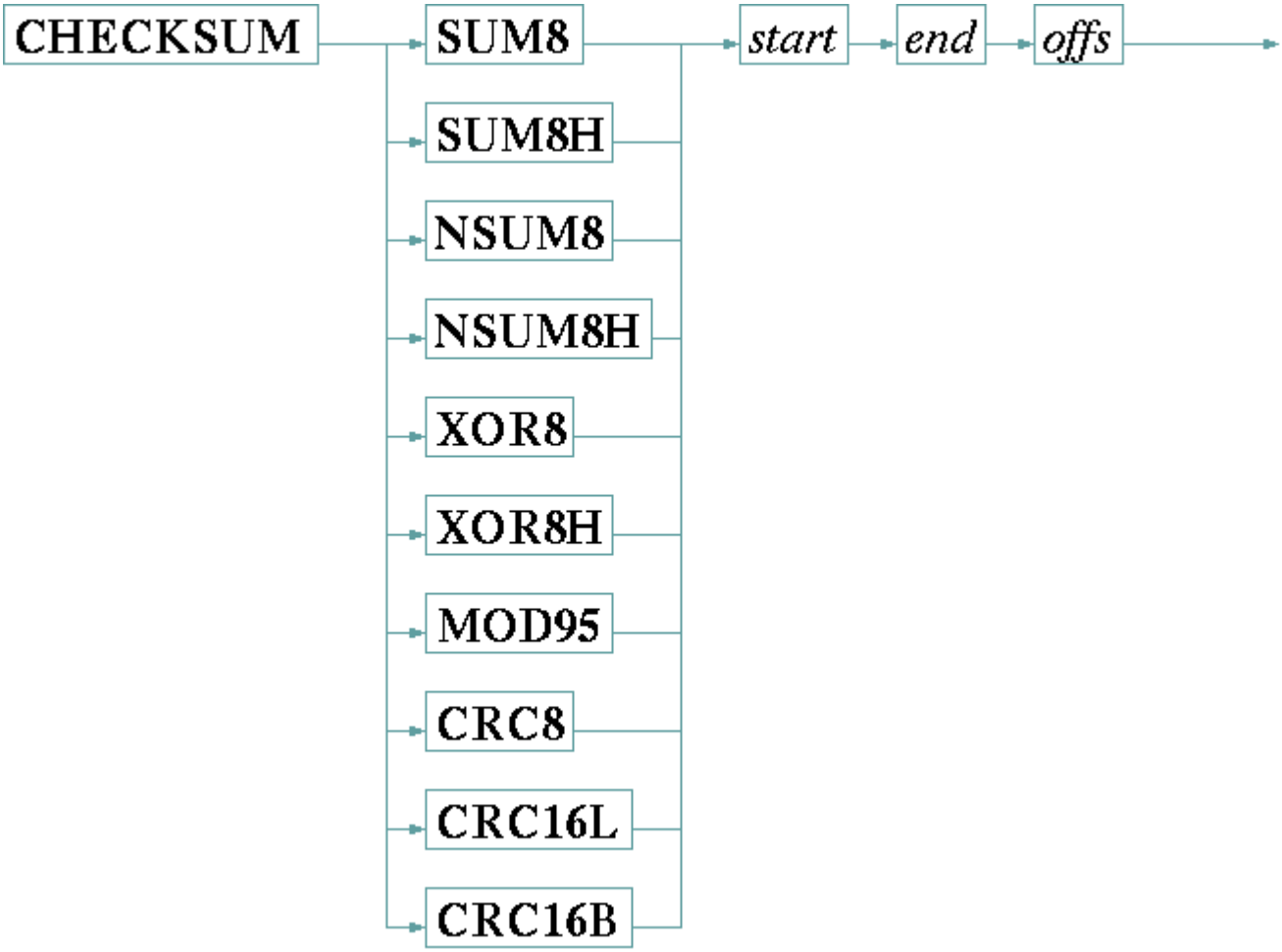
The terminating character may be specified either as a single character enclosed in double quoted or as a decimal number.

The `offset` parameter defines how many characters still have to be read or how many characters have to be cut at the end of the data to isolate the user data part of the message:

- If *offset = 0* the data read, including the termination character, is returned as the user data of the message.
- If *offset < 0* the given number of characters are cut off at the end of the data. E.g. *offset = -1* removes the termination character from the data.
- If *offset > 0* the given number of characters additionally is read and appended to the user data.

7.6.5.6 CHECKSUM

The `CHECKSUM` step reads a checksum and compares it to the checksum calculated from the message contents. If the values differ, the reception of this message is aborted and a communication fault is reported.



The parameters of this step specifier are:

SUM8	Expects a one byte checksum done by adding all bytes and truncating the sum to the lowest 8 bits.
SUM8H	Like `SUM8`, but expects the checksum coded as a 2 digit hex number.
NSUM8	Expects a one byte checksum done by adding all bytes, taking the negative value of the sum and then truncating the sum to the lowest 8 bits.
NSUM8H	Like `NSUM8`, but expects the checksum coded as a 2 digit hex number.
XOR8	Expects a one byte checksum, done by XOR-ing all specified bytes.
XOR8H	Like `XOR8`, but expects the checksum coded as a 2 digit hex number.
MOD95	Expects the one character 'modulo 95' checksum, known from the protocol used by Miteq devices (see below).
CRC8	Expects a one byte `CRC8` checksum.

CRC16L	Expects a two byte `CRC16` checksum, calculated with the ARC/LHA CRC algorithm. Expects the least significant byte first (little endian).
CRC16B	Expects a two byte `CRC16` checksum, calculated with the ARC/LHA CRC algorithm. Expects the most significant byte first (big endian).
start	The buffer position of the first character to be included in the checksum computation (0 means the first character).
end	The buffer position of the last character to be included in the checksum computation, relative to the actual position (`-1` means the last character read).
offs	The buffer position where the checksum starts, relative to the actual position (`-1` means the last character read).

The `MOD95` type checksum is computed following the formula:

$$Checksum = 32 + MOD_{95}\left[\left(\sum_{i=1}^{N} message\ byte_i\right) - (32 * N)\right]$$

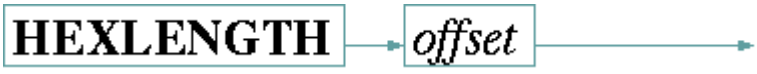
7.6.5.7 DATALENGTH

The `DATALENGTH` protocol step reads a one byte binary packet length field.

The received data length is remembered by the protocol until aUSERDATA step is encountered. The USERDATA step will read this number of bytes *minus* the offset stated with the `DATALENGTH` step.

7.6.5.8 HEXLENGTH

The DATALENGTH protocol step reads packet length field coded as a two byte hexadecimal number.



The received data length is remembered by the protocol until aUSERDATA step is encountered. The USERDATA step will read this number of bytes *minus* the offset stated with the HEXLENGTH step.

7.7 Device oriented user interface

The ***sat-nms*** software provides a so called *device oriented* user interface for each type of equipment it supports. If you double click to a device icon at the MNC user interface you

automatically get a device window for this device. The software selects the right type of device window for the device driver which is configured for this device.

To use a device oriented user interface for the device drivers you added to the software by yourself, it is necessary to extend the framework which implements the device oriented user interface. The device oriented user interface for a particular device consists of one or more parameter screens which are made specially for this device type and a number of *universal* screens which appear with almost any device. So, there are principally two steps to do:

- Define the required parameter screens for the new device type
- Define the set of screens which shall be selectable in the device window.

7.7.1 How the software finds the screens for a device

To extend the device oriented user interface, it is important to understand how device drivers and the predefined user interface windows are linked together in the ***sat-nms*** software. Principally there are three steps from the device driver to the user interface:

1. Each device driver defines a variable called `info.frame` . The device driver sets this value to the name of the frame definition it expects.
2. The user interface software - when the operator clicks to a device icon - reads the frame definition file of this name in the `dframes` subdirectory which is usually located on standard installation in `/home/satnms/dframes` .
3. The frame definition file contains the list of screens (the names of the files defining the screen layouts). The user interface software builds a card folder like window from this information.

7.7.2 Creating new screens

The software keeps the files containing the screen layouts for the device oriented user interface in a directory called `dscreens` which is usually located on standard installation in `/home/satnms/dscreens` . There is one file for each parameter screen. The screens are created with the same tool used to design customer supplied screens for the task oriented user interface.

To protect the predefined screens from being accidentally modified, the layout editing tool normally has no access to these screens. To edit a screen for the device oriented user interface, run the client with the following command line options:

Linux / on the MNC server

```
java -cp client.jar satnms3.guiconf.Configurator localhost /dscreens/...
```

where `...` is the name of the screen to edit. This opens the layout editing tool with the given file name. Please note, that the MNC server must be running at this time. The layout editing tool reads the file via the data base capabilities of the server rather than accessing the disk file directly.

Linux / on a remote machine You also may start the layout editing tool on a client computer over the LAN. At this place open a terminal sessionand enter:

```
java -cp client.jar satnms3.guiconf.Configurator [ip] /dscreens/...
```

Like above, **...** has to be replaced by the name of the screen to edit. `[ip]` stands for the IP address of the MNC server.

Windows / on a remote machine You also may start the layout editing tool on a MS Windows based client computer over the LAN. At this place open a command prompt window or create a shortcut with the following command line:

```
java -cp client.jar satnms3.guiconf.Configurator [ip] /dscreens/...**
```

Like above, `...` has to be replaced by the name of the screen to edit. `[ip]` stands for the IP address of the MNC server. If your computer does not recognize the `java`` command, try **jre** instead and check the installation of your Java Runtime Environment.

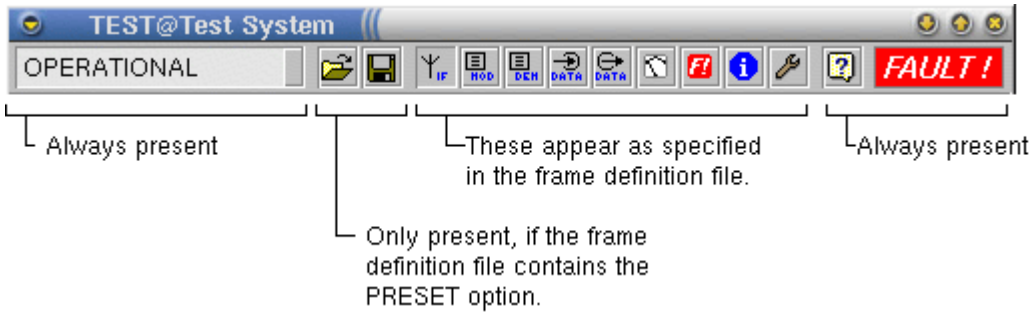
Editing the screen layout for the parameter screens is straight forward. It is recommended to use a layout similar to the existing screens. The software's user manual describes the usage of the layout editing program and introduces the types of display elements a screen may consist of.

7.7.3 Creating a 'frame' definition

The set of screens making up the window for a particular device type is called a frame. Frames are defined by text files located in the `~/dframes` directory of the software installation. To add a new frame type to the software, you have to edit such a file.

```
              PRESETS
tb-if.gif;       /dscreens/Modem-General;         General / IF parameters
tb-mod.gif;      /dscreens/Modem-TX;              Modulator parameters
tb-dem.gif;      /dscreens/Modem-RX;              Demodulator parameters
tb-dinput.gif;   /dscreens/Modem-TX-IFC;          TX interface parameters
tb-doutput.gif;  /dscreens/Modem-RX-IFC;          RX interface parameters
tb-meter.gif;    /dscreens/Modem-Measure;         Meter Readings
tb-fault.gif;    @FAULTS;                      Faults and fault mask
tb-info.gif;     @INFO;                       Device Info
tb-tool.gif;     @CONFIG;                      Maintenance
```

The example above shows a frame definition file for a SCPC satellite modem. Lines starting with a '#' are comments, they are ignored when the file is read. Below the button bar of the device window resulting from this frame definition is shown:



- The operation mode selector on the left and the help button and fault mark at the right are contained in all device screens, they need not to be specified in the frame definition file.
- The `PRESET` keyword in the third line of the frame definition file tells the software to display the buttons for parameter preset storage and retrieval in the button bar of the device window. The preset load / save buttons by default require the operator to be logged in with privilege level 100 or higher. You may append a number to the `PRESET` keyword, separated by a blanc character in order to specify the minimum privilege level for load / save preset buttons in this device window explicitly. `PRESET 150` for example makes the buttons requires privilege level 150 or higher.

The table starting at line five of the file specifies the other buttons in the tool bar and the screens bound to them. The buttons appear in the same order as the lines of the table. Each table row consists of three fields, separated by semicolon characters:

- The **first** entry selects the icon to be displayed. The following chapter shows the icons available with their file names.
- The **second** entry specifies the screen that shall be shown on a click to this button. This either is the name of a screen file (usually from the `dscreens` directory) or one of these keywords selecting a special screen:
  - **@FAULTS** : Shows the common screen listing the faults of this device.
  - **@INFO** : Shows the common screen listing the info variables for this device.
  - **@CONFIG** : Shows the common maintenance and configuration screen for this device.
  - **@SATLIST** : Shows a screen interfacing to an antenna pointing database (used by the AntennaPointing logical device)
- The **third** field defines the the help text for this button which is shown above the mouse cursor if this id hold above the button.

All device frames provided by SatService contain the @FAULTS, @INFO and @CONFIG screens in the way shown in the example.

7.7.4 Icon reference

column 1	column 2	column 3	column 4

column 1	column 2	column 3	column 4
tb-1.gif	tb-123.gif	tb-2.gif	tb-3.gif
tb-4.gif	tb-5.gif	tb-6.gif	tb-a1.gif
tb-abc.gif	tb-accept.gif	tb-ackmail.gif	tb-ainput.gif
tb-aoutput.gif	tb-apsdvb.gif	tb-apsskywan.gif	tb-asiout.gif
tb-async.gif	tb-audio1.gif	tb-audio2.gif	tb-audio3.gif
tb-audio4.gif	tb-audio5.gif	tb-audio6.gif	tb-audio7.gif
tb-audio8.gif	tb-aupc.gif	tb-avoutput.gif	tb-az.gif
tb-back.gif	tb-backup.gif	tb-c2n.gif	tb-cam.gif
tb-cgroup.gif	tb-chain.gif	tb-chanlist.gif	tb-chart.gif
tb-chdir.gif	tb-checkmark.gif	tb-chhist.gif	tb-clock.gif
tb-colors.gif	tb-compare.gif	tb-condacc.gif	tb-connect.gif
tb-connflt.gif	tb-copy.gif	tb-cross.gif	tb-cut.gif
tb-data.gif	tb-decode.gif	tb-delete.gif	tb-dem.gif
tb-device.gif	tb-devices.gif	tb-devicex.gif	tb-dflash.gif
tb-dial.gif	tb-dialing.gif	tb-diallist.gif	tb-dinout.gif
tb-dinput.gif	tb-dirup.gif	tb-disconnect.gif	tb-doutput.gif
tb-downarrow.gif	tb-draw.gif	tb-dst.gif	tb-dvbs2.gif
tb-el.gif	tb-encode.gif	tb-erasor.gif	tb-exclam.gif
tb-exec.gif	tb-eye.gif	tb-farray0.gif	tb-farray1.gif
tb-farray2.gif	tb-farray3.gif	tb-farray4.gif	tb-farray5.gif
tb-farray6.gif	tb-farray7.gif	tb-farray8.gif	tb-fastdown.gif
tb-fastfwd.gif	tb-fastrew.gif	tb-fastup.gif	tb-fault.gif
tb-faultlist.gif	tb-faultlog.gif	tb-filexfer.gif	tb-frame.gif

column 1	column 2	column 3	column 4
tb-frm.gif	tb-ftrack.gif	tb-geopos.gif	tb-gflash.gif
tb-go.gif	tb-gref.gif	tb-grid.gif	tb-groups.gif
tb-hangup.gif	tb-help.gif	tb-ibs.gif	tb-idr.gif
tb-if.gif	tb-image.gif	tb-info.gif	tb-input.gif
tb-ipinout.gif	tb-leftarrow.gif	tb-line.gif	tb-link.gif
tb-linklist.gif	tb-list.gif	tb-livebelt.gif	tb-livelog.gif
tb-lnb.gif	tb-lock.gif	tb-locked.gif	tb-login.gif
tb-magnify.gif	tb-mail.gif	tb-master.gif	tb-measlist.gif
tb-meter.gif	tb-mod.gif	tb-modems.gif	tb-move.gif
tb-mpegout.gif	tb-mux.gif	tb-new.gif	tb-newchild.gif
tb-newdev.gif	tb-newdevs.gif	tb-newifc.gif	tb-newitem.gif
tb-newnode.gif	tb-nosound.gif	tb-now.gif	tb-nowarn.gif
tb-onestep.gif	tb-open.gif	tb-output.gif	tb-paste.gif
tb-pause.gif	tb-play.gif	tb-plug.gif	tb-pmirror.gif
tb-pmlchk.gif	tb-pmrchk.gif	tb-preset.gif	tb-print.gif
tb-properties.gif	tb-question.gif	tb-queue.gif	tb-record.gif
tb-rectangle.gif	tb-red.gif	tb-redial.gif	tb-redstop.gif
tb-relay.gif	tb-reload.gif	tb-restore.gif	tb-rf.gif
tb-rflash.gif	tb-rightarrow.gif	tb-rmail.gif	tb-route.gif
tb-rs.gif	tb-rx.gif	tb-sainteract.gif	tb-sat-a.gif
tb-sat-c.gif	tb-sat-m.gif	tb-sat-opt.gif	tb-sat-p.gif
tb-sat-s.gif	tb-sat-st.gif	tb-sat1.gif	tb-sat2.gif
tb-sat3.gif	tb-sat4.gif	tb-sat5.gif	tb-satellite.gif
tb-satout.gif	tb-save.gif	tb-saveas.gif	tb-savevlc.gif
tb-schedule.gif	tb-search.gif	tb-send.gif	tb-setdev.gif

column 1	column 2	column 3	column 4
tb-sigma.gif	tb-slave.gif	tb-slavelink.gif	tb-smodems.gif
tb-sound.gif	tb-spectrum.gif	tb-spider.gif	tb-src.gif
tb-srcdst.gif	tb-stop.gif	tb-switch.gif	tb-swoff.gif
tb-swon.gif	tb-text.gif	tb-textend.gif	tb-tmove.gif
tb-tool.gif	tb-tools.gif	tb-transmitting.gif	tb-trash.gif
tb-tree.gif	tb-treeedit.gif	tb-tsanalyze.gif	tb-tsoutput.gif
tb-tx.gif	tb-undo.gif	tb-unlocked.gif	tb-uparrow.gif
tb-vgroup.gif	tb-vinput.gif	tb-vlc.gif	tb-vlclist.gif
tb-vlcmap.gif	tb-voptim.gif	tb-voutput.gif	tb-vsearch.gif
tb-vupdate.gif	tb-wizard.gif	tb-xzoomfull.gif	tb-xzoomin.gif
tb-xzoomout.gif	tb-yflash.gif	tb-ystar.png	tb-yzoomfull.gif
tb-yzoomin.gif	tb-yzoomout.gif	tb-zoomin.gif	tb-zoomout.gif

7.8 Online Help

The ***sat-nms*** software provides a context sensitive online help based on HTML files. To make this help system easy to maintain, a simple type setting language has been defined for it. The source of the help system is contained in a couple of Markdown files containing plain text and some formatting commands. A help compiler creates all the HTML files for the online help from the sources.

With the ***sat-nms*** software there come beside a ready compiled online help also the help compiler and all sources of the help system. This enables you to extend the online help system for your needs.

You however should be aware, that the original help files (sources) are restored with each software update you install at the system. To avoid conflicts with the documentation supplied by SatService, you should modify or extend the online help only in the following situation:

1. You have written your own device driver, and you want to supply a help page for this type of equipment.

7.8.1 Help file format

Help files are now simply defined in Markdown a lightweight markup language for creating formatted text using a plain-text editor and are stored in `.md` files.

We are supporting the standard Markdown as described athttps://commonmark.org/.

If you want to have your own help files included into the online help system, please provide the new driver definition and your help files to SatService GmbH and we will automatically build the new online help with your additional files.

Attention! The old ***sat-nms*** help file format was replaced by this standardized Markdown files.

For your reference please find below this deprecated information:

So the following chapter describes the format of the source files the online manual of the software is built from.

Topic Definitions

The online manual divides the complete text into topics, where each topic is compiles to one web page. A topic is identified by it's unique four digit topic number. This number is used to build the file name of the web page and also to reference the page in hyper links.

In the source text a topic starts with a topic definition line and ends where the next topic starts. The topic definition looks like this:

.topic TTTT L The title of this topic

The keyword '.topic' must start at the very first column of the line. Further on, TTTT is the for topic number of the topic to define. L is a number in the range 1..4 defining the level of this heading in the table of contents. The remaining part of the line defines the title of the topic.

Paragraphs

The HTML text is shown by the browser as left justified text, made up to the width of the browser window. Line breaks and paragraphs do **not** appear where you type them in the source code, you have explicitly to insert line break or paragraph break commands into the source text to get these breaks in the output.

.p	Closes a paragraph. Most browsers show this as a line break followed by a half height blank line.
.br	Inserts a line break.

Both commands may appear anywhere in the line, however, usually they are placed either at the line end or in the own line.

Formatted Lists

You may typeset ordered (numbered) lists and unordered (bullet) lists in the text. Browsers usually display each list item as a paragraph, preceded by either the paragraph number or a

list item symbol. You may nest lists of both types to any depth. The help compiler simply translates the list formatting commands to HTML, it does not check if each list is properly terminated. If you produce invalid HTML code because of missing list end marks, some browsers will give strange results.

.ul	Starts a bullet list.
.ol	Starts an ordered (numbered) list
.li	Starts a list item.
.br	Inserts a line break.
.eul	Closes a bullet list.
.eol	Closes an ordered (numbered) list

Although these commands are recognized everywhere in the line, formatting lists as shown in the example below makes the source well readable.

```
.ol
  .li This is the first list item
  .li This is the second one, it contains a lot of text which does not fit in one line.
  .li The third item is a short one again.
.eol
```

Tables

Help pages may contain tables consisting of an almost arbitrary number of columns. Tables appear with a pale blue background. You have no control over the precise layout of the table, it is shown as the browser likes to display plain HTML tables with a cell background set. Because no geometry must be specified, it is very simple to define the table in the source text:

.tl	Starts a table definition
.ts	Starts the first cell of a table row
.tc	Separates a table cell from the next one in the same row
.te	Marks the end of the last table cell in a row.
.etl	Closes the table definition.

With the commands shown above, the table gets defined cell by cell, row by row. Table definitions are translated by the help compiler one by one to HTML tags. When defining a table, be careful to define the same number of cells in each row and to terminate each row properly by a **.te** command. The example below shows in which way a simple table definition should be

formatted in the source:

```
.tl
.ts cell 1/1   .tc cell 1/2   .tc cell 1/3   .te
.ts cell 2/1   .tc cell 2/2   .tc cell 2/3   .te
.ts cell 3/1   .tc cell 3/2   .tc cell 3/3   .te
.etl
```

Images / Pictures

The user manual also contains images and diagrams. Files in GIF or JPG format may be included. The image files must be located in the same directory as the other files of the online help. You find already a bunch of image files there, most of them are copies of the tool bar icons used by the software.

.i file-name	Includes an image.

If this is the only command in the line, be sure to add a single space character after the file name. This is due to a flaw of the help compiler which does not recognize the file name properly if it ends at the line end.

Pre-formatted Text

The text may include segments of 'pre-formatted' text, which the browser displays using a mono-spaced font with line breaks at the same positions as in the source. Pre-formatted text is used to include for example excerpts of configuration files.

.pre	Starts pre-formatted text
.epre	Ends pre-formatted text

The `.pre` and `.epre` commands must appear as single commands in a line, starting at the first column.

Emphasizing Text

Portions of text may be *emphasized* , **bold printed** or set in a `typewriter font` . To do this, enclose the text to highlight in curly braces and insert a letter `e` for emphasized, `b` for bold or `t` for typewrite directly after the opening brace. There must no be a space character between the opening brace and the code character. The opening and closing braces must be in the same line, nesting different hi-lighting styles is not allowed.

Cross-references / Hyper-links

You may include HTML hyper-links into the text to permit the reader to jump directly from one topic to another by a single mouse click. To insert a hyper-link, To do this, enclose the text to

hi-light in braces and insert the sequence `h####` directly after the opening brace. `####` is to be replaced by the topic number to link to. Like with the character formatting codes above, there must no be a space character between the opening brace and the `h`. The opening and closing braces must be in the same line, no nesting of braces is allowed.

Paragraph Headings

There is no special command to insert paragraph headings. The topic title is the only heading for a help page. Only this will appear in the table of contents in the printed version.

To introduce headings on a paragraph level, use a bold printed, one line paragraph instead.

Escaping Dot Commands

Words in the help text containing dots (e.g. file names) may be confused with a dot command sequence by the help compiler. Example: `MyDevice.proto` in the source file will be printed as `MyDevice` at the end of one paragraph and `roto` at the beginning of the next one. This is because the `.p` sequence in the file name is translated to a paragraph break.

The `.dot` sequence is used to escape the dot command in such a situation. Spelling the file name `MyDevice.proto` in the source produces the desired output.

Comments

You may add comments or annotations to the source text of the document which later are not shown in the manual. Lines which start with a double percent character `%%` at the first column are skipped when the file is processed by the help compiler.

Conditional Compilation

The help compiler implements a mechanism to include or skip parts of the source text depending on a couple of environment information here called *features*. The conditional compilation mechanism closely follows that one used with common programming language compilers:

%if feature	Marks the start of the documents part which only shall be included if 'feature' is present with this software installation
%else	Marks the end of `%if` branch, starts another one with the negated condition. The `%else` statement is optional
%endif	Marks the end of this `%if` or `%if` / `%else` clause.

The `%if` , `%else` and `%endif` statements must appear singly in a line and must start at the first column in the line.

The following 'features' may be tested in an `%if` clause (testing other features, always results in *not present*):

7.8.2 Rebuilding the online help

The `help` directory of the ***sat-nms*** software installation contains a shell script called makehelp. If you change to the `help` directory and call this script there, the complete online help system gets rebuilt. You should do this in one of the following cases:

- You have added a device driver to the software.
- You have edited the help file for a new device driver.
- You have edited the 'local.hlp' file.
- You installed a software update.

7.8.3 Adding help files for new devices

To display a context sensitive help for each device driver, the help system uses a separate file for each each device type, containing some information how to use and to configure the driver/device. This help topic is called if you click to the help button in a device's window.

Device driver help files use a slightly different format than the main help file does. The difference is, the a device driver help file must not contain topic definitions. One topic definition for each file is generated automatically when the help system is built.

Device driver help files are located in the `help` directory of the ***sat-nms*** software installation. The files are named like the drivers itself, but with the extension `.hlp` . The help compiler creates dummy files for missing device driver files.

The help files for the device drivers supplied by SatService all follow a common format. They first explain for which type of equipment this driver was designed for. The the button bar of the device window is described in a table. Below this the configuration/setup parameters for this device driver are explained, some device specific remarks may follow. You are encouraged to use this scheme for your customer supplied device drivers. too.

7.9 ServiceClient library

The ***sat-nms*** ServiceClient application lets you easily define virtual devices and group them to *services* which show a subset of important parameters of these devices and a summarized fault state of all devices of a service.

The ServiceClient library defines the device types the ServiceClient knows and the implementation of these devices types by ***sat-nms*** device drivers.

The ServiceClient library is an XML file which is read by the ServiceClient application at startup. You may modify this file to make the ServiceClient recognize additional ***sat-nms*** device drivers as implementation of a predefined device type or you even may define additional device types from scratch and add some implementations to them. Beside the device type definitions the XML file contains some settings which influence the GUI appearance of the ServiceClient.

The following chapters describe how to add new device types and implementations to the library and how to adapt GUI settings like the colors used for signalling the fault state of devices and services. Later chapters then describe the file format of the XML file in detail.

7.9.1 Adding new devices or device types to the library

To make the ***sat-nms*** ServiceClient recognize devices of a new type, there must be at least one `implementation` for the device type defined in the

- Devices containing multiple instances
- Fault detection
- Conditional implementation

7.9.2 Adapting the appearance of the ServiceClient

Some aspects of the ServiceClient's GUI appearance are defined in a section of the `definitions.xml` file. These are beside others the colors used to signal states like `OK`, `WARNING` or `FAULT` in the GUI. You may change these settings by editing the `definitions.xml` file, with the next start of the ServiceClient the changed settings will be used.

Chapter GUI constants later in this document gives a complet list of GUI constants which may be adapted by editing this file.

7.9.3 File format of definitions.xml

7.9.4 Device type definitions

7.9.4.1 Parameters

7.9.4.2 Implementations

7.9.5 GUI constants

The `definitions.xml` file contains a section `<gui ... </gui>` which defines a number for constant definitions which let you adapt some aspects of the program's appearance to your needs. Every definition in this sections has the form:

```
<def name="..." value="..." />
```

For each definition, `name` is a unique name for the constant, `value` is the value assigned to it. All definitions have default values which apply if the definition is not found.

Below, a table shows all known constant definitions with a description of their meaning and the default values the software assumes if the definition is not found in the `definitions.xml` file.

name	type	default	description

name	type	default	description
editorPrivilegeLevel	integer	150	The minimum privilege level an operator must have in order to edit the service configuration. For operators below this privilege level the toolbar button for the edit mode is without function.
leftPaneWidth	integer	250	The width (pixels) of the "service" screen elements which make up the left screen pane. Usually there is no need to change this except are dealing with very ling service names or comment strings which do not fit in the default width.
rightPaneWidth	integer	600	The width (pixels) of the screen elements in the right screen pane which show the parameters of a device in the service. The width of these screen elements does not automatically grow with the number of parameters shown in it, so it may be necessary to enlarge this if you are going to define new device types with many parameter in one row. The default width is sufficient to display three "wide" and one "narrow" parameters in a single row.
okBackgroundColor	color	00FF00 (green)	The background color used to signal that an element is in OK state. The service widgets on the left pane, the header of the service detailled view on the right pane and the fault indicator of each device are using this color definition. The color is expressed as a 6-digit hexadecimal RGB value defining the red, green and blue component of the color.

name	type	default	description
warningBackgroundColor	color	FFFF00 (yellow)	The background color used to signal that an element is in WARNING state. The service widgets on the left pane, the header of the service detailled view on the right pane and the fault indicator of each device are using this color definition. The color is expressed as a 6-digit hexadecimal RGB value.
faultBackgroundColor	color	FF0000 (red)	The background color used to signal that an element is in FAULT state. The service widgets on the left pane, the header of the service detailled view on the right pane and the fault indicator of each device are using this color definition. The color is expressed as a 6-digit hexadecimal RGB value.
okForegroundColor	color	000000 (black)	The text / foreground color used to signal that an element is in OK state. If you change the "okBackgroundColor", you may want to adapt the color used to the foreground as well to maintain sufficient contrast
warningForegroundColor	color	000000 (black)	The text / foreground color used to signal that an element is in WARNING state. If you change the "okBackgroundColor", you may want to adapt the color used to the foreground as well to maintain sufficient contrast
faultForegroundColor	color	FFFFFF (white)	The text / foreground color used to signal that an element is in FAULT state. If you change the "okBackgroundColor", you may want to adapt the color used to the foreground as well to maintain sufficient contrast

If you want to revert a value to its default and do not know the default setting, we recommend

to change the 'def' tag into someting other (e.g. usedefault) instead of deleting the line defining this definition. This way the name of the definition is still in the file, but due to the modified tag it will not be read.

7.10 Reference Tables

This chapter provides a number of tables with reference information which may be useful as a quick lookup for questions regarding the extension of the ***sat-nms*** software. In particular, the supplied tables are:

- Device driver keyword reference
- Protocol definition keyword reference
- Help file keyword reference
- Debugging with terminal session
- Global faults

7.10.1 Device driver keyword reference

The following table lists the keywords recognized in a device driver specification file in alphabetical order. When viewing this table online you may use the hyper-links in the description column to jump to the place where this keyword is explained in the manual.

=	Part of the syntax of a SET or BITSET statement.
ALARM	The ALARM statement defines a faults flag.
AT	Within an INPUT statement, the AT clause moves the read cursor to a certain position in the received data.
BIGENDIAN	Specifies that a WRITE or READ statement shall treat multibyte integer numbers in big endian (MSB first) byte order.
BIT	Reads a single bit from an integer number within an INPUT statement.
BITMERGE	Should no longer be used.
BITS	Specifies to write or read a number of bits with in a certain byte in a WRITE or a READ statement.
BITSET	The BITSET statement isolates a single bit from a numeric variable and assigns it to another one.
BITSPLIT	Should no longer be used.
BOOL	Defines a variable to be a boolean flag in a VAR statement.
CALL	The CALL statement calls a procedure as a subroutine.

CBER	Interprets a two character string as a bit error rate within an INPUT statement.
CHOICE	Defines a variable to be a choice list in a VAR statement.
CHOICES	Modifies a choice list variable with the RANGESET statement.
CLASS	The CLASS statement defines the Java class which does the real work. Customer supplied device drivers do not require this statement.
COMMENT	The COMMENT statement defines an identification for the driver which is reported to the user interface.
COPY	The COPY statement copies a single parameter or a complete device setup from one device to another.
CUT	Cuts a number of characters in the INPUT statement.
CYCLE	Defines the refresh cycle for a variable in the VAR statement.
DELAY	The DELAY statement pauses the driver execution for a certain time.
DISABLED	Marks a variable to be initially disabled in a VAR statement or disables a variable with the RANGESET statement.
DRATE	The DRATE statement computes an interface data rate from a symbol rate and modulation/encoding parameters.
ENABLED	Enables a variable with the RANGESET statement.
FLOAT	Defines a variable to be a floating point number in a VAR statement.
FLOAT16	Writes or reads a 16 bit floating point number with a WRITE or READ statement.
FLOAT32	Writes or reads a 32 bit floating point number with a WRITE or READ statement.
FLOAT64	Writes or reads a 64 bit floating point number with a WRITE or READ statement.
FMT	Formats a number to a string in the PRINT statement.
GET	Marks a procedure to be a GET-type one in the PROC statement.
GOTO	The GOTO statement jumps to a label in a procedure.

HEX	Interprets a string as a hexadecimal number in an INPUT statement or defines a variable to be a hexadecimal number in a VAR statement.
IF	The IF statement conditionally executes the following statement.
INCLUDE	The INCLUDE statement reads another source file.
INIT	Tells the driver to initialize a variable at power up with a certain value in the VAR statement.
INPUT	The INPUT statement reads a message from the device and parses the reply as text.
INT16	Writes or reads a 16 bit integer number with a WRITE or READ statement.
INT32	Writes or reads a 32 bit integer number with a WRITE or READ statement.
INT64	Writes or reads a 64 bit integer number with a WRITE or READ statement.
INT8	Writes or reads an 8 bit integer number with a WRITE or READ statement.
INTEGER	Defines a variable to be an integer number in a VAR statement.
INVALIDATE	The INVALIDATE statement marks a variable to be re-read in this or the next driver cycle.
LOG	The LOG statement writes a message into the event log.
MAX	Modifies the upper range limit of a numeric variable with the RANGESET statement.
MIN	Modifies the lower range limit of a numeric variable with the RANGESET statement.
OBJECT	Defines a variable to be a 'object' in a VAR statement.
OFFSET	Adds an offset to as value in a PRINT or an INPUT statement.
PRINT	The PRINT statement composes a text message/command and sends this to the device.
PROC	The PROC keyword starts a procedure definition.
PROTOCOL	The PROTOCOL statement defines the (preferred) communication protocol a driver is designed to use.

PROTOCOLPARAMETER	The PROTOCOLPARAMETER statement defines parameters to be passed to the communication protocol.
PUT	Marks a procedure to be a PUT-type one in thePROC statement.
PUTPRIORITY	Defines that the driver shall use the put priority execution scheme for PUT/GET procedures. See Topic Tweaking the processing sequence for details.
RAISECOMMFAULT	The RAISECOMMFAULT statement makes the driver to raise a communication fault.
RANGESET	The RANGESET statement modifies range properties of a variable during runtime.
READ	The READ statement gets a message from the device and parses the data as a binary data structure.
READHEX	The READHEX reads a message from the device and formats the data as a hex dump string.
READONLY	Marks a variable to be read only in aVAR statement or makes the variable read only with the RANGESET statement.
READWRITE	Makes a variable writable with the RANGESET statement.
SAVE	Tells the driver do save a variable's value on disk in theVAR statement.
SCALE	Multiplies a value with a scale factor in a PRINT or an INPUT statement.
SEND	The SEND statement sends a parameter message to another device.
SET	The SET statement assigns a value to a variable.
SETUP	Makes a variable appear in the setup menu for the device VAR statement.
SKIP	Skips whitespace characters in the INPUT statement.
SRATE	The SRATE statement computes a symbol rate from a data rate and modulation/encoding parameters.
SUBROUTINE	Marks a procedure to be a subroutine when following the PROC keyword.
SUBSCRIBE	Marks a procedure to be called if the value of a variable changes when following the PROC keyword.

SYNC	Syncs a variable, updates the value to be commanded with the value recently read back from the device.
TABLE	The TABLE statement defines a translation table.
TEXT	Defines a variable to be plain text in aVAR statement.
TRM	Cuts a string up to a given termination character in theINPUT statement.
VAR	The VAR statement defines a driver variable.
WATCH	Binds a procedure to one or more variables with thePROC statement.
WRITE	The WRITE statement composes a binary message/command and sends this to the device.
WRITEHEX	The WRITEHEX statement converts a hex dump string to a binary message, sends this to the device.
XLT	Translates a character string through a table in aPRINT or an INPUT statement.

7.10.2 Protocol definition keyword reference

The following table lists the keywords recognized in protocol definition files in alphabetic order.

ADDRESS	Outputs the device address.
ADDRESS	Reads the device address.
CHAR	Outputs a single character.
CHAR	Reads a single character.
CHECKSUM	Outputs a message checksum.
CHECKSUM	Reads a message checksum.
CLASS	Specifies the underlying protocol class.
COMMENT	Defines the protocol identification string.
DATALENGTH	Outputs a data length field (binary).
DATALENGTH	Reads a data length field (binary).
HEXLENGTH	Outputs a data length field (hex).
HEXLENGTH	Reads a data length field (hex).

PROTOCOLPARAMETER	Defines parameter to be passed to the communication protocol
RECEIVE	Starts the receive protocol step specification.
SEQUENCE	Outputs a binary message sequence number.
START	Reads a message start character.
STRING	Reads a terminated string as the user data
TRANSMIT	Starts the transmit protocol step specification.
USERDATA	Outputs the user data.
USERDATA	Reads the user data.

7.10.3 Help file keyword reference

Attention! Please remember that this is deprecated and the new file format ist Markdown.

deprecated:

The following table lists the 'dot commands' recognized by the help compiler in alphabetic order. They all are explained in the chapter ' Help file format '.

.ul	Starts a bullet list.
.ol	Starts an ordered (numbered) list
.li	Starts a list item.
.br	Inserts a line break.
.eul	Closes a bullet list.
.eol	Closes an ordered (numbered) list
.topic	Defines a topic header.
.p	Inserts a paragraph break.
.br	Inserts a line break.
.dot	Inserts a '.' (dot) character.
.tl	Starts a table definition
.ts	Starts the first cell of a table row
.tc	Separates a table cell from the next one in the same row

.te	Marks the end of the last table cell in a row.
.etl	Closes the table definition.
.i	Includes an image.
.pre	Starts pre-formatted text
.epre	Ends pre-formatted text

7.10.4 Debugging with Terminal Session

The ***sat-nms*** client provides the *terminal session window* which opens a TCP connection to Port 2000 of the MNC server. This debugging port provides several command to intercept devices and device communication and to manipulate parameters. It is also possible to access this port with any terminal program (e.g. putty with connection type `raw``).

7.10.4.1 Terminal Session Commands

Enter `?` on the command prompt to get the following short help overview:

```
---> ?
general information:
  server / faults / msg-count
messages:
  get <msg-id> / set <msg-id> <v> / loop <msg-id> <v>
message distributors:
  cdis (debug|nodebug)
  ddis (debug|nodebug)
device threads:
  threads / thread <n>
client peers:
  peers / peer <n> (view|verbose|silent|kill)
system:
  restart / shutdown / q
debug log:
  limit / nolimit
```

server

Shows information of ***sat-nms*** MNC server process including:

- state: current server state. R = normal operation
- clients: number of connected clients
- d-threads: number of devices threads (see `threads`)
- msg-rate: The `server` command outputs the current rate of updated driver variables. The rate is determined over an interval of 5 seconds. Variable update calls are counted and divided by the time that has passed since the rate was last determined. The value is

a bit too high in terms of communication to real equipment, because also the (internal) messages between logical devices are counted as well

Example:

```
---> server
state:      R
clients:    3
d-threads:  54
msg-rate:   505.69 msgs/sec
```

faults

Shows a list of all devices (one per line) and the current summary faults or warning state. Devices which are ok, just show no message text.

Example: where `HPA-2` is in WARNING state and `HPA-3` in FAULT state

```
...
HPA-1:
HPA-2: WARNING
HPA-3: Summary FAULT
S1:
S2
```

msg-count

Outputs the absolute counter readings of the messages distributed by the MNC server since startup. Starts with a high number, because all the range messages are counted at the start.

Example:

```
distributed messages
  from devices 2167806180
  to devices   2261258
```

cdis debug / cdis nodebug

Enables or disabled debug output on stdout (usually found in `/home/satnms/.panic.log`) for the client distributor. This show messages going from server to the clients (including other devices).

Attention : Enabling this output creates huge amount of date which on the disk. Do not forget to disable after testing.

Example: A beacon level (parameterId `BCRX.power`) is send to client and to a File-Recorder logical device.

```
BCRX.power distributed to satnms3.vlc.ClientPeer (176)
BCRX.power distributed to satnms3.device.FileRecorder@57576068
```

ddis debug / ddis nodebug

Enables or disabled debug output on stdout (usually found in `/home/satnms/.panic.log`) for the device distributor. This show messages going to devices (including logical devices).

Attention : Enabling this output creates huge amount of date which on the disk. Do not forget to disable after testing.

Example: A BUC is commanded to transmit (parameterId `BUC-1.tx.on`). This messages usually have their origin from user input.

Example:

```
BUC-1.tx.on distributed to tx.on ON
```

threads

Shows all list of all threads (which corresponds to the interface definition) and the current thread state. A threads / interface can have 1 or more attached devices.

Example:

```
---> threads
0   I  DT-null     (2 devices)
1   R  DT-null     (1 devices)
2   I  DT-null     (6 devices)
3   I  DT-10.10.1.13:4001(1 devices)
4   I  DT-ttyS11    (1 devices)
5   I  DT-ttyS0     (1 devices)
```

thread `n`

Show details for thread `n` . This includes (for each device attached to this thread/interface)

- device name
- device class: a Java class name if the device is hardcoded or contains hardcoded function, or satnms3.device.UniversalDriver if the driver is written completely in the ***sat-nms*** Universal Device Driver Language
- driver name
- measured cycle time: minimal/average/maximal measured cycle time in msec. This show how long the device needs to go through all GET (and PUT) procedures. It includes the idle time as configured for this interace (usually 1000 msec)
- protocol name
- protocol class: a Java class name if the protocol is hardcoded or satnms3.proto.TTYProtocol if the protocol is written in the ***sat-nms*** Universal Device

Driver Language
- interface
- response time: minimal/average/maximal measured response time for messages send to a (real) device in msec.
- thread state

Example: ***sat-nms*** LBRX Beacon Receiver with HTTP protocol

```
---> thread 19
thread 19 DT-null (I)
  device 1
    device name: BCRX-1M8
    device class: satnms3.device.UniversalDriver
    driver name: SatService-Beacon-Receiver 3.03 200430
    measured cycle time: 1124/1254/1378 msec
    protocol name: HTTP 1.01 090728
    protocol class: satnms3.proto.HTTProtocol
    interface: null
    response times: 120/157/215 msec
```

peers

Show list of all connected clients and their current state

Example: 3 connected clients from different hosts

```
---> peers
0   RI  10.10.1.143
1   RI  192.168.2.22
2   RI  192.168.2.34
```

peer `n` (view|verbose|silent|kill)

peer function	description
verbose	print all message between client and server
silent	disable verbose output
kill	close connection, drop client
view	show peer details

Example: shows details for client 1

```
---> peer 0 view
ClientPeer 0:
id:             50
destination:    10.10.1.143
input state:    R
output state:   I
queue fill:     0
uptime:         16d21h
msgs received:  297387
msgs sent:      2637717
```

Client peer input and output states:

state	description
I	Initializing
R	Reading / waiting for Data
P	Processing Data
D	Dead (Streams closed)
O	Outputting (sending Data)
S	Stopping

Example: enabled client/server verbose output where a new beacon level is send to the client and a keep alive message is exchanged between client and server (ping)

```
sent BCRX.power -71.56
sent @ping. Fri Jul 30 15:09:47 UTC 2021
got  @cpng.
```

restart

restart MNC server, similar to `satnms-stop; satnms-start` on command line or the restart button the client UI.

shutdown

shutdown MNC server. similar to to `satnms-stop` on command line

q

close terminal session

limit / nolimit

disabled log rotation of `/home/satnms/debug.log` .

Attention : Enabling this output creates huge amount of date which on the disk which will cleaned up automatically. Do not forget to disable after testing.

7.10.4.2 Device Message Commands

Within the terminal session it is possible to read and write every parameter of all devices. A typical use case is to mock parameters of equipment which is currently not connected.

command	description
get `msg-id`	show the current value of a parameter specified by `msg-id`
set `msg-id` `v`	set the value `v` of parameter specified by `msg-id`. This injects a message to the event distributor in the same way as when a user command a value from a client
loop `msg-id` `v`	set the value `v` of parameter specified by `msg-id`. This injects a message to the event distributor that looks like it was received from the equipment (response on polling)

get `msg-id`

The get command returns all parameters starting with the string in `msg-id`. If you specify a name which does not exist it returns an empty line. If `msg-id` matches more than one parameter multiple lines will be return. This allows for example to get all parameter from one device with `get BUC-1` or a subset of parameters like `get BUC-1.faults`.

Normally 2 lines return per parameters:

- the parameter itself with the current value
- the range setting of this parameter (marked with `[msg-id].R`) which describes the type range and range.
- Disabled parameters (marked with `DIS`) or parameters which where never initialized will only result in one line with the range message.

Example: Parameter BUC-1.tx.on which is a CHOICE parameter with possible values ON and OFF, currently set to ON

```
---> get BUC-1.tx.on
BUC-1.tx.on.R EnumRange (ON|OFF)
BUC-1.tx.on OFF
```

The get command recognizes a parameter `-r` which causes the command to suppress all range messages which otherwise would be included in the list.

```
---> get -r BUC-1.tx.on
BUC-1.tx.on OFF
```

The range message belonging to this paramter is not shown.

set `msg-id` `value`

The 'set' command does not produce any output (except logParameterChanges is enabled for the device). It performs also no range checking and setting values of an unknown messageId will be silently ignored.

In case of a communication fault, the driver will stop after the first INPUT statement which does not get answered by the equipment and starts from the beginning. Therefore you will not see any outgoing (PUT) commands even if you set a parameter with `set` until you stop the driver cycle aka polling (see below).

Example: switch BUC transmit to ON

```
---> set BUC-1.tx.on OFF
000000 2021-07-30 16:07:50 2021-07-30 16:07:50   I BUC-1 tx.on changed to 'OFF' by debug
```

loop `msg-id` `value`

A loop commands simulates that the device receives a message from equipment. So it looks like, that the equipment response to polling. With this command you can mock e.g. read-only values to display something useful even if you have no equipment connected.

But remember, values set with the `loop` command will be overwritten by next successful polling result (if there is communication to the equipment)

Example: set the output power of an BUC to 162 (Watt). On first `get` BUC-1.meas.fwdPwr is not initialized, but after `loop` ing 162 to the device it show this new value on the next `get`

```
---> get BUC-1.meas.fwdPwr
BUC-1.meas.fwdPwr.R IntegerRange R/O (0 .. 0)

---> loop BUC-1.meas.fwdPwr 162

---> get BUC-1.meas.fwdPwr
BUC-1.meas.fwdPwr 162
BUC-1.meas.fwdPwr.R IntegerRange R/O (0 .. 0)
```

verbose mode

Enable *verbose mode* for a device to follow all communication between MNC server and equipment *Verbose mode* could be enabled in the device setup page of each device or in the terminal session with `set [DEVICENAME].verbose true`.

Example: Beacon Receiver with HTTP protocol. Parameter BCRX.threshold is read from the device with the command `/rmt?thrh=?`

```
---> set BCRX.verbose true
tx(BCRX): /rmt?thrh=?
rx(BCRX): thrh=-90.00
  BCRX.threshold=-90.00
```

In addition you can enable *dump mode* for a device to print full communication between MNC's server and equipment even if it does not match with the required protocol structure with `set [DEVICENAME].dump true`.

Attention : Enabling this output creates huge amount of date which on the disk so do not forget to disable after testing.

stop polling cycle

You can stop the driver cycle with the following command in the terminal session: `set [DEVICENAME].cycle true`. Re-enable polling with `set [DEVICENAME].cycle false` or a restart of the MNC server.

Example: disable polling and set a value on a not connected device. After disabling polling the device tries one more time to communicate (`tx(BUC-1): GET ...`) but does not expect a response anymore. Device state switches to Communication OK.

```
---> set BUC-1.cycle false
tx(BUC-1): GET 1.3.6.1.4.1.2566.127.1.5.10.1.1.14.1.2.0 (FAILED)
000000 2021-07-30 15:52:20 2021-07-30 15:52:20   I BUC-1 past 0m17s Communication OK.
000000 2021-07-30 15:52:20 2021-07-30 15:52:20   I BUC-1 OK.
```

Now you can issue a `set` command (or send commands via Client UI) and the corresponding PUT procedures will be executed. Of course you will still not get any response from the equipment, so the Communication Fault will be issued.

```
---> set BUC-1.tx.on OFF
000000 2021-07-30 15:57:46 2021-07-30 15:57:46   I BUC-1 tx.on changed to 'OFF' by debug
tx(BUC-1): SET 1.3.6.1.4.1.2566.127.1.5.10.1.2.1.3.1.0 I 1 (FAILED)
000000 2021-07-30 15:57:55 2021-07-30 15:57:55   I BUC-1 past 2m57s Communication OK.
000000 2021-07-30 15:57:55 2021-07-30 15:57:55   I BUC-1 OK.
```

7.10.5 Global faults

The M&C software knows two global faults flags indicating fault conditions which are not linked do devices or subbsystems:

SYSTEM.fault

indicates some sort of general fault, mostly faults occurring during the initialization phase of the M&X. SYSTEM.fault may be one of "OK." and "Summary FAULT". Which condition caused

the fault is listed in the event log. A "RESET" to the SYSTEM device clears this fault.

SUMMARY.fault

summarizes the the faults of all devices in the M&C. SUMMARY.fault knows two states, 'FAULT' and 'OK.'. It turns to 'FAULT' if at least one device in the M&C shows a fault.

8 API Reference

This document describes the sat-nms WebGUI backend server API which is used by the Web front end of the software and also may be used to control a sat-nms M&C in a programmatic way.

Quick access links to the main sections of the document:

- HTTPS Communication and OAuth2 Authentication
- API End Points
- WebSocket Communication
- Data Models
- Satellite Database

8.1 HTTPS Communication and OAuth2 Authentication

The sat-nms WebGUI backend server uses HTTPS for all communication. Unsecured HTTP calls are not accepted. In most cases a non-standard port number is used by the server, details about the port and also about the certificate provided by the server may vary for every installation of the software.

The backend server uses OAuth2 for authorization. It may be configured to accept JWTs issued by an external provider. In parallel the backend server provides a service for obtaining and renewing access tokens as well.

8.2 API End Points

This section describes the API end points provided by the sat-nms WebGUI backend server. The end point descriptions appear in an order which groups related functions. An alphabetic list of all end points is shown below.

- /api/v1/activebackend
- /api/v1/acutargets
- /api/v1/antennas
- /api/v1/applypreset
- /api/v1/backendinfo
- /api/v1/channels
- /api/v1/collectpreset
- /api/v1/dbstate
- /api/v1/dbversion
- /api/v1/debug
- /api/v1/devicesetup
- /api/v1/dframes
- /api/v1/documents
- /api/v1/drivers
- /api/v1/dscreens

- /api/v1/eventack
- /api/v1/eventreport
- /api/v1/filerecordings
- /api/v1/filerecordersettings
- /api/v1/globalSettings
- /api/v1/healthcheck
- /api/v1/inventory
- /api/v1/logout
- /api/v1/macros
- /api/v1/makewebsocket
- /api/v1/mnclist
- /api/v1/peek
- /api/v1/playmacro
- /api/v1/poke
- /api/v1/preferredprotocols
- /api/v1/presets
- /api/v1/presetvars
- /api/v1/devicevars
- /api/v1/protocols
- /api/v1/redundancy
- /api/v1/restart
- /api/v1/satdetails
- /api/v1/satellites
- /api/v1/satoperators
- /api/v1/schedule
- /api/v1/streamkeys
- /api/v1/subscribe
- /api/v1/tleimport
- /api/v1/thumbnail
- /api/v1/trackmem
- /api/v1/treeview
- /api/v1/uscreens
- /api/v1/user
- /api/v1/users
- /api/v1/userSettings
- /healthcheck
- /token

8.2.1 /token

The /token API call requests an access token from the server which can be used to authorize subsequent API calls.

All API calls except /token require a valid access token (JWT) to be passed with the header of the request as a "Authorization: Bearer ..." header or the server will abort with a 401 NOT AUTHENTICATED error.

Supported HTTP methods: POST

POST /token

Requests an access token for the user specified with the parameters added to this call. The parameter to this request must be passed as HTML form style parameters (application/x-www-form-urlencoded). The same end point is used to get a new and to renew an existing token.

Parameters:

Name	Value
username	The name of the user to log in. Must be a valid user name known to the sat-nms M&C application.
password	The password supplied by the user for login (clear text).
grant_type	One of 'password' or 'refresh_token'
refresh_token	A valid refresh token, making the backend to reply with a new version of the existing access token linked to this refresh token.

The *grant_type* parameter is mandatory, if it is set to 'password' the parameters *password* and *username* are requires as well. If *grant_type* is set to 'refresh_token', the parameter *refresh_token* is mandatory as well.

On success the API call returns a TokenReply document containing beside other information the access token and a refresh token to renew the access token.

Return Codes:

Code	Description
200	OK, login was successful. A TokenReply document is replied.
400	*grant_type* is neiter 'password' nor 'refresh_token'
401	Login failed. In this case a ApiError document is returned, describing details of the error. This happens either with *grant_type*='password' and invalid credentials or with *grant_type*='refresh_token' and an invalid refresh token.

8.2.2 /api/v1/logout

Logs out the currently logged in user.

Supported HTTP methods: POST

POST /api/v1/logout

Logs out the currently logged in user. Requires a valid access token to be passed with the request header to identify the session to be terminated. In fact, this call does not logout all

open sessions of a given user, it terminates exactly one session which is referenced by the token.

Parameters: none

Return Codes:

Code	Description
204	OK, logout was successful. No payload data follows.

8.2.3 /healthcheck

This API call can be used to check if the backend server responds to API calls. It works without being logged in.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /healthcheck

Returns an Dictionary document listing some basic status information about the backend server. As this API call is with public access, it returns only a limited set of information.

The Dictionary document returned contains the following keys with its values:

Key	Description
PRODUCT	The name of this product
WEBSITE	The SatService website URL
VERSION	The software version actually running
SERIALNO	The software serial number

Parameters: none

Return Codes:

Code	Description
200	OK, the Dictionary with the status information is returned as payload.

8.2.4 /api/v1/user

Delivers some information about the user currently logged in. Returns aUserProperties document describing the user. This document contains -- beside other information -- the

privilege level of the user which must be known to decide which user interface elements must be shown in a read-only way because the user is not authorized to change these parameter (but he may read them if logged in).

This API end point also may be used with PUT to change the password of the actually logged in user.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/user

Returns a UserProperties document describing the currently logged in user.

Expected Payload: none

Replied Payload: a UserProperties document with information about the user who issued this call.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserProperties document describing the currently logged in user. If no information about the currently logged in user is available, a dummy Document describing a user with the name 'unknown' and the privilege level '0' is returned.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/user

Changes the password of the user issuing this call. Returns the (updated)UserProperties for the currently logged in user.

Expected Payload: a UserProperties document containing the new password for the actual user. All other fields of the the document are ignored.

Replied Payload: a UserProperties document with the updated user information.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserProperties with the updated data for the actually logged in user.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.

8.2.5 /api/v1/users

Reads or writes the list of users known to the backend. When editing the user list, the list must be read with a GET and later sent back with a PUT with all fields in each UserProperties record.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/users

Gets a UserList document containing the list of all known users.

Expected Payload: none

Replied Payload: a UserList document containing the list of all known users.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a UserList containing the list of all known users.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/users

Send a UserList document containing the list of all known users to the backend. The UserList replaces the previous one. Already logged in users remain logged in, even if they are deleted from the list or if their password was changed.

Expected Payload: The UserList document to write.

Replied Payload: The UserList document echoed back.

Parameters: none

Return Codes:

Code	Description
200	OK, returns the written UserList echoed back.

Code	Description
400	cannot change the password. Probably the user logged in by LDAP. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

8.2.6 /api/v1/mnclist

Delivers a list containing some information about the M&C systems managed by the backend. Returns a MncList document containing this list.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/mnclist

Returns a MncList document containing the list of managed M&C systems and their properties.

Parameters: none

Return Codes:

Code	Description
200	OK, returns a MncList document containing the list of managed M&C systems.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

8.2.7 /api/v1/activebackend

Delivers an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching. Also permits to change this using a PUT call.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/activebackend

Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Parameters: none

Expected Payload: none.

Replied Payload: Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Return Codes:

Code	Description
200	OK, returns the expected document
401	not logged in. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/activebackend

Sets the name of the of the backend which is actually considered to be active for M&C redundancy switching.

Parameters: none

Expected Payload: An ActiveBackend document containing the name of the backend which shall be considered to be active for M&C redundancy switching.

Replied Payload: Returns an ActiveBackend document showing the name of the backend which is actually considered to be active for M&C redundancy switching.

Return Codes:

Code	Description
200	OK, the name of the active backend has been updated. Returns the expected document
401	not logged in. In this case a ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

8.2.8 /api/v1/peek

Reads ('peeks') the Parameter or Range definition addressed by {messageId} from the backend's message cache. If no such message is present in the cache, the M&C server is asked to deliver the message. The API call waits up to 3 seconds for the M&C server to answer, then a 404 error is returned. Peeking parameters is a very inefficient way to get a parameter value. It only should be used if a parameter value is required only once (without subsequent updates) or for testing purposes. Section WebSocket Communication describes the subscription model using STOMP/websockets which is the preferred way to read/update parameters in the UI.

The {messageId} supplied to /api/v1/peek must be a complete message identifier consisting of M&C-, device- and parameter-ID.

The document returned on success is either a Parameter or a Range, depending on the {messageId} supplied. For details about messages and message identifiers, see section

Message later in this document.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/peek/{messageId}

Returns a Message document describing the requested parameter.

Parameters:

Parameter	Description
messageId	The requested messageId. Must be a complete message identifier consisting of M&C-, device- and parameter-ID.

Return Codes:

Code	Description
200	OK, returns a Message document describing the requested parameter.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no parameter with the given messageId exists. In this case a ApiError document is returned, describing details of the error.

8.2.9 /api/v1/poke

Writes ('pokes') a Parameter or Range message to the M&C server. The message to be written must be supplied as POST data.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/poke

Expects exactly one Parameter document as POST data. This creates and distributes the message on the M&C server addressed by the M&C identifier contained in the supplied message. This does not check if the specified destination exists. Messages sent to non-existing destinations are silently ignored, also result in a 201 response.

Parameters: none

Return Codes:

Code	Description
201	Success, echoes the message created.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.

8.2.10 /api/v1/makewebsocket

Creates a new websocket to connect to. A unique ID is generated for this websocket and returned as a WebsocketId document in the call's response body.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/makewebsocket

Creates a new websocket to connect to. A unique ID is generated for this websocket and returned as a WebsocketId document in the call's response body.

The WebsocketId is required to open the socket itself and for all subscribe / unsubscribe operations on this websocket. See section Websocket Communication for details about how to open a websocket.

Parameters: none

Return Codes:

Code	Description
201	Success, delivers the ID of the created websocket in the response body (data model WebsocketId).
401	not logged in. In this case a ApiError document is returned, describing details of the error.

8.2.11 /api/v1/subscribe

Subscribes or unsubscribes the delivery of one or more message IDs over the websocket connection. Whether to subscribe or unsubscribe is defined by the HTTP method.

Supported HTTP methods: POST, DELETE, HEAD, OPTIONS

POST /api/v1/subscribe/{websocketId}/{messageId}

Subscribes for the reception of messages with the given messageId over the websocket identified by websocketId. The requested message and all subsequent updates are sent over the websocket connection.

Returns a SubscribeList with the response body containing the message IDs which have been successfully subscribed (in this case of a single subscription the list contains either one or no message IDs).

The API call does not check if the remote M&C system can deliver a message with the given ID. Messages may become available later, this is a common situation.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to send the message and subsequent updates to. See section Websocket Communication how to obtain a websocketID and how to open a websocket.
messageId	Identifies the message / parameter to subscribe. Must be a complete sat-nms message identifier including the M&C ID.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the (single) subscribed message ID.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

POST /api/v1/subscribe/{websocketId}

Subscribes for the reception of messages of multiple message IDs over a websocket. This variant of the API call expects a SubscribeList data object in the request body. This data object contains the message IDs to subscribe for.

The subscription requests are forwarded and are checked if successful one by one. The API call returns a SubscribeList in the response body containing the message IDs which have been successfully subscribed.

The API call does not check if the remote M&C system can deliver messages with the given IDs. Messages may become available later, this is a common situation.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to send the message and subsequent updates to. See section Websocket Communication how to obtain a websocketID and how to open a websocket.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the subscribed message IDs.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

DELETE /api/v1/subscribe/{websocketId}/{messageId}

Unsubscribes the reception of messages with the given messageId over the websocket identified by websocketId.

Returns a SubscribeList with the response body containing the message IDs which have been successfully unsubscribed (in this case the list contains either one or no message IDs).

The API call does not check if there is a subscription pending for the given parameters. If a subscription is canceled which never has been done or which has been unsubscribed before, this is silently ignored.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to unsubscribe. See section Websocket Communication how to obtain a websocketID and how to open a websocket.
messageId	Identifies the message / parameter to unsubscribe. Must be a complete sat-nms message identifier including the M&C ID.

Return Codes:

Code	Description
200	Success, a SubscribeList is returned with the response body containing the (single) unsubscribed message ID.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

DELETE /api/v1/subscribe/{websocketId}

Unsubscribes the reception of multiple message IDs over a websocket. This variant of the API call expects a SubscribeList data object in the request body. This data object contains the message IDs to unsubscribe.

The subscription requests are forwarded and are checked if successful one by one. The API call returns a SubscribeList in the response body containing the message IDs which have been successfully subscribed.

The API call does not check if there are subscriptions pending for the given message IDs. If a subscription is canceled which never has been done or which has been unsubscribed before, this is silently ignored.

Parameters:

Parameter	Description
websocketID	Identifies the websocket to unsubscribe. See section Websocket Communication how to obtain a websocketID and how to open a websocket.

Return Codes:

Code	Description
201	Success, a SubscribeList is returned with the response body containing the unsubscribed message IDs.
401	Not logged in. In this case a ApiError document is returned, describing details of the error.
403	Invalid websocketId. The websocketId referenced in the URL either does not exist or the session which sent the request does not own this websocket. An empty SubscribeList is returned with the response body.

8.2.12 /api/v1/uscreens

Gives access to the user defined screen definitions stored on the primary M&C. Screen definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/uscreens

Returns a list of all available uscreens.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available user screens. The list is read from the primary M&C.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available user screens.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/uscreens/{name}

Returns a ScreenDefinition document describing the requested screen. The screen definition is read from the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to load. Please note, that screen names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a ScreenDefinition document describing the requested screen.

Return Codes:

Code	Description
200	OK, returns a ScreenDefinition document describing the requested screen.
400	the primary M&C is offline, the screen cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
404	no screen definition with the given name exists. In this case a ApiError document is returned, describing details of the error.

POST /api/v1/uscreens/{name}

Stores a ScreenDefinition at the primary M&C. If the screen already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the screen to store. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The ScreenDefinition to store.

Replied Payload: Echoes back the written ScreenDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written ScreenDefinition.
400	the primary M&C is offline, the screen cannot be written. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/uscreens/{name}

Deletes a ScreenDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to delete. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the screen has been deleted.
400	the primary M&C is offline, the screen cannot be deleted. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	not found. No screen of this name exists. In this case aApiError document is returned, describing details of the error.

8.2.13 /api/v1/dscreens

Gives access to the device screen definitions stored on the primary M&C. Screen definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/dscreens

Returns a list of all available dscreens. The list is read from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available device screens.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available device screens.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/dscreens/{name}

Returns a ScreenDefinition document describing the requested screen. The screen is read from the primary M&C.

Parameters:

Parameter	Description

Parameter	Description
name	The name of the screen to load. Please note, that screen names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a ScreenDefinition document describing the requested screen.

Return Codes:

Code	Description
200	OK, returns a ScreenDefinition document describing the requested screen.
400	the primary M&C is offline, the screen cannot be read. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no screen definition with the given name exists. In this case a ApiError document is returned, describing details of the error.

POST /api/v1/dscreens/{name}

Stores a ScreenDefinition at the primary M&C. If the screen already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the screen to store. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The ScreenDefinition to store.

Replied Payload: Echoes back the written ScreenDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written ScreenDefinition.
400	the primary M&C is offline, the screen cannot be written. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/dscreens/{name}

Deletes a ScreenDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the screen to delete. Screen definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the screen has been deleted.
400	the primary M&C is offline, the screen cannot be deleted. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	not found. No screen of this name exists. In this case aApiError document is returned, describing details of the error.

8.2.14 /api/v1/dframes

Gives access to the device frame definitions stored on the primary M&C. Device frame definitions may be listed, read or written with this call.

Supported HTTP methods: GET, POST, HEAD, OPTIONS

GET /api/v1/dframes

Returns a list of all available device frame definitions. The list is read from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: Returns an ItemList containing the names of all available device frame definitions.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the names of all available device frame definitions.
400	the primary M&C is offline, the list cannot be read. In this case aApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

GET /api/v1/dframes/{name}

Returns a DeviceFrameDefinition document describing the requested frame. The frame definition is read from the primary M&C.

Parameters:

Parameter	Description
name	The name of the frame definition to load. Please note, that frame names are case sensitive, the name must match exactly to be loaded.

Expected Payload: none.

Replied Payload: Returns a DeviceFrameDefinition document describing the requested frame.

Return Codes:

Code	Description
200	OK, returns a DeviceFrameDefinition document describing the requested frame.
400	the primary M&C is offline, the frame definition cannot be read. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no frame definition with the given name exists. In this case aApiError document is returned, describing details of the error.

POST /api/v1/dframes/{name}

Stores a DeviceFrameDefinition at the primary M&C. If the frame definition already exists, it gets overwritten.

Parameters:

Parameter	Description
name	The name of the frame to store. DeviceFrame definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: The DeviceFrameDefinition to store.

Replied Payload: Echoes back the written DeviceFrameDefinition.

Return Codes:

Code	Description
200	OK, echoes back the written DeviceFrameDefinition.
400	the primary M&C is offline, the frame definition cannot be written. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/dframe/{name}

Deletes a DeviceFrameDefinition at the primary M&C.

Parameters:

Parameter	Description
name	The name of the frame definition do delete. DeviceFrame definition names may consist of the characters A-Z, a-z, 0-9 and the hyphen character.

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the device frame has been deleted.
400	the primary M&C is offline, the frame definition cannot be deleted. In this case a ApiError document is returned, describing details of the error.
401	not logged in. In this case a ApiError document is returned, describing details of the error.

Code	Description
404	not found. No device frame of this name exists. In this case aApiError document is returned, describing details of the error.

8.2.15 /api/v1/macros

Permits to load and save sat-nms macro definitions as well as to read the list of macros defined on a given M&C.

Supported HTTP methods: GET, PUT, DELETE, HEAD, OPTIONS

GET /api/v1/macros/{mncName}

Returns an ItemList document containing the list of macros defined on the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the macro list shall be loaded.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the requested macro list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case aApiError document is returned, describing details of the error.

GET /api/v1/macros/{mncName}/{macroName}

Returns a Macro document containing the requested macro.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the macro shall be loaded.
macroName	The name of the macro to load.

Return Codes:

Code	Description
200	OK, returns a Macro containing the requested macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/macros/{mncName}/{macroName}

Saves a macro on the given M&C. Expects Macro document containing the macro text to save as request body. The *name* field in the Macro document will be ignored, the macro will be saved under the name given in the URL. As every PUT request, this will either overwrite an existing macro with this name or create a new macro definition at the M&C if no macro with this name exists.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the macro shall be saved.
macroName	The name of the macro to save.

Return Codes:

Code	Description
200	OK, returns the Macro containing the saved macro. This will have the *name* field set the name under which the macro actually has been saved.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/macros/{mncName}/{macroName}

Deletes a macro at the given M&C.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C on which the macro shall be deleted.
macroName	The name of the macro to delete.

Return Codes:

Code	Description
200	OK, returns the Macro containing the deleted macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

8.2.16 /api/v1/playmacro

Executes a macro on a given M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/playmacro/{mncName}

Executes the referenced macro. Expects the macro to play as aMacro body, delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the macro shall be loaded and executed.

Return Codes:

Code	Description
200	OK, returns the Macro containing the executed macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	the M&C with the given name does nit exist. In this case aApiError document is returned, describing details of the error.

POST /api/v1/playmacro/{mncName}/{macroName}

Executes the referenced macro. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the macro shall be loaded and executed.
macroName	The name of the macro to execute.

Return Codes:

Code	Description
200	OK, returns the Macro containing the executed macro.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no macro with the requested name. In this case a ApiError document is returned, describing details of the error.

8.2.17 /api/v1/presets

Permits to load and save sat-nms device preset definitions as well as to read the list of presets defined on a given M&C.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/presets/{mncName}

Returns an ItemList document containing the list of preset directories found on the given M&C. This is the list of device drivers which have at least one preset saved on this M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset directory list shall be loaded.

Return Codes:

Code	Description

Code	Description
200	OK, returns an ItemList containing the requested preset directory list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

If the given M&C name exists, but there are no presets stored at all, the request returns 200 OK and an empty list.

GET /api/v1/presets/{mncName}/{driverName}

Returns an ItemList document containing the list of presets defined this device driver on the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset list shall be loaded.
driverName	The name of the device driver for which the preset list shall be loaded.

Return Codes:

Code	Description
200	OK, returns an ItemList containing the requested preset list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

If the given M&C name exists, but there are no presets stored for the driver type, the request returns 200 OK and an empty list.

GET /api/v1/presets/{mncName}/{driverName}/{presetName}

Returns a DevicePreset document containing the requested preset.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C from which the preset shall be loaded.
driverName	The name of the device driver for which the preset shall be loaded.
presetName	The name of the preset to load.

Return Codes:

Code	Description
200	OK, returns a DevicePreset containing the requested preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no preset with the requested driver/name combination. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/presets/{mncName}/{driverName}/{presetName}

Saves a preset on the given M&C. Expects DevicePreset document containing the preset text to save as request body. The *name* and *driver* fields in the DevicePreset document will be ignored, the preset will be saved under the driver / name given in the URL. As every PUT request, this will either overwrite an existing preset with this name or create a new preset definition at the M&C if no preset with this name exists.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the preset shall be saved.
driverName	The name of the device driver for which the preset shall be saved.
presetName	The name of the preset to save.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the saved preset. This will have the *name* and *driver* fields set the values under which the preset actually has been saved.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

DELETE /api/v1/presets/{mncName}/{driverName}/{presetName}

Deletes a preset from the given M&C. Expects no request body.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the preset shall be deleted.
driverName	The name of the device driver for which the preset shall be deleted.
presetName	The name of the preset to delete.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the deleted preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists or no device preset with that name / device driver could be found. In this case a ApiError document is returned, describing details of the error.

8.2.18 /api/v1/applypreset

Applies a preset on a given M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/applypreset/{mncName}/{deviceName}

Applies a preset to the given device. Expects a DevicePreset as request body, then delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be applied.
deviceName	The name of the device to apply the preset to.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the applied preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

POST /api/v1/applypreset/{mncName}/{driverName}/{deviceName}/{presetName}

Applies the referenced preset to the given device. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be applied.
driverName	The name of the device driver for which the preset shall be applied.
deviceName	The name of the device to apply the preset to.
presetName	The name of the preset to apply.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the applied preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no driver/device/preset with the requested name. In this case a ApiError document is returned, describing details of the error.

8.2.19 /api/v1/collectpreset

Collects a preset on a given M&C. This tells the referenced device to collect its settings into a preset and to store this preset in the M&C with the given name.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/collectpreset/{mncName}/{driverName}/{deviceName}/{presetName}

Collects a preset on a given M&C. This tells the referenced device to collect its settings into a preset and to store this preset in the M&C with the given name. Expects no request body, simply delegates the request to the M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C at which the preset shall be collected.
driverName	The name of the device driver for which the preset shall be collected.
deviceName	The name of the device to collect the preset from.
presetName	The name of the preset to store.

Return Codes:

Code	Description
200	OK, returns the DevicePreset containing the collected preset.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either M&C with the given name exists or at the known M&C there exists no device with the requested name. In this case a ApiError document is returned, describing details of the error.

8.2.20 /api/v1/presetvars

Delivers the list of driver variable which may be stored in a preset for a given device driver.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/vi/presetvars/{mncName}/{driverName}

Returns a PresetVars document containing the requested list of preset-storable variables. The backend reads the device driver *driverName.device* from the given M&C, compiles the driver text and selects the preset storable variables from the compiled list. The returned PresetVars

document contains the Range definitions for all variables of the driver which may appear in a device preset. The *messageId* field in each Range object is set to the name of the variable.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the variable list shall be loaded.
driverName	The name of the device driver for which the variable list shall be evaluated.

Return Codes:

Code	Description
200	OK, returns a PresetVars document containing the requested variable list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either no M&C with the given name exists, or at the known M&C there exists no device driver with the given name. In this case a ApiError document is returned, describing details of the error.

8.2.21 /api/v1/devicevars

Delivers the list of all variables a device defines

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/vi/devicevars/{mncName}/{deviceName}

Returns a DeviceVars document containing the requested list of variables. The backend evaluates the driver for the given device, compiled the device driver and gets the variable list from this. The returned DeviceVars document contains the Range definitions for all variables of the driver which may appear in a device preset. The *messageId* field in each Range object is set to the name of the variable.

The API call returns the variable list as it exists at compile time at system startup. Changes made dynamically to the list of variables or to some variable ranges are not contained in the list

Parameters:

Parameter	Description
mncName	The name of the M&C from which the variable list shall be loaded.

Parameter	Description
deviceName	The name of the device for which the variable list shall be evaluated.

Return Codes:

Code	Description
200	OK, returns a PresetVars document containing the requested variable list.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	either no M&C with the given name exists, or at the known M&C there exists no device driver with the given name. In this case a ApiError document is returned, describing details of the error.

8.2.22 /api/v1/schedule

Permits to load and save the macro scheduler's schedule at a given M&C.

When subscribing for a '.SYSTEM.schedule' parameter of a M&C, the front end receives a message which indicates 'an updated schedule is available' instead of sending the bulky schedule itself over the websocket connection.

After the front end received this message, it can use the /api/v1/schedule API call to retrieve the schedule as a Schedule data object.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/schedule/{mncName}

Returns a Schedule document containing the actual system schedule from this M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C from which the schedule shall be loaded.

Return Codes:

Code	Description
200	OK, returns a Schedule document containing the requested schedule.

Code	Description
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

PUT /api/v1/schedule/{mncName}

Overwrites the system schedule at the given M&C. Expects a Schedule document containing new schedule to set.

Parameters:

Parameter	Description
mncName	The name of the M&C on which the preset shall be saved.

Return Codes:

Code	Description
200	OK, returns the Schedule containing the saved schedule.
401	not logged in. In this case a ApiError document is returned, describing details of the error.
404	no M&C with the given name exists. In this case a ApiError document is returned, describing details of the error.

8.2.23 /api/v1/healthcheck

This API call delivers some status information about the running backend server. Unlike the version with public access, this version of the health check contains the complete set of information.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/healthcheck

Returns an Dictionary document listing some status information about the backend server.

The Dictionary document returned contains the following keys with its values:

Key	Description
PRODUCT	The name of this product

Key	Description
WEBSITE	The SatService website URL
VERSION	The software version actually running
SERIALNO	The software serial number
UPTIME	The server uptime
MEMORY	The memory usage of the backend
SESSIONS	The number of open sessions
WEBSOCKS	The number of open (message) websocket connections

Parameters: none

Return Codes:

Code	Description
200	OK, the Dictionary with the status information is returned as payload.

8.2.24 /api/v1/dbstate

This API call delivers some status information about the backend's connection to the database server. It may be used by a frontend to evaluate if the backend actually is connected to the (read-only) backeup database and therefore no changes to the database content are allowed

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/dbstate

Returns a SatDbState document listing some status information about the database connection.

Parameters: none

Return Codes:

Code	Description
200	OK, the SatDbState document with the status information is returned as payload.

8.2.25 /api/v1/dbversion

This API call delivers the version information about the database connected to the backend.

Supported HTTP methods: GET,HEAD,OPTIONS

GET /api/v1/dbversion

Returns a SatDbVersion document showing the actual version of the database and the version required by the backend.

Parameters: none

Return Codes:

Code	Description
200	OK, the SatDbState document with the status information is returned as payload.

8.2.26 /api/v1/satellites

This API endpoint gives access to various functions of the sat-nms satellite database, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS**GET /api/v1/satellites**

Returns the list of stored satellites as an array ofSatDbSatellite data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored satellites as an array ofSatDbSatellite data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all satellites and their attributes.
401	not logged in. In this case anApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}

Returns the basic properties of the satellite addressed by {satelliteId} as aSatDbSatellite data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The requested SatDbSatellite data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites

Creates a new satellite with the given attributes . The unique *satelliteId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbSatellite data object to store. Any *satelliteId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbSatellite data object added to the database. The replied data object contains the *satelliteId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbSatellite data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}

Updates an existing satellite with the given *satelliteId* from the attached SatDbSatellite data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbSatellite data object containing the fields to be updated in the database. If the data object contains a *satelliteId* key, the value is ignored. The database record is solely identified by *satelliteId* set in the URL, the *satelliteId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbSatellite data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}

Remove a satellite from the database. This deletes the satellite's record from the satellite table and all records from other tables like beacons, TLEs etc which unambiguously depend on the deleted satellite.

Parameters:

Parameter	Description
satelliteId	Id of the satellite to delete

Expected Payload: none

Replied Payload: The SatDbSatellite data object which has been deleted.

Return Codes:

Code	Description
200	OK, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/positions

Returns the list of satellite positions for a given satellite as an array of SatDbPosition data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite positions as an array of SatDbPosition data objects. This array may be empty if there are no positions defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/beacons

Returns the list of satellite beacons for a given satellite as an array of SatDbBeacon data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite beacons as an array ofSatDbBeacon data objects. This array may be empty if there are no beacons defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Returns the information stored for a particular satellite beacon addressed by {satelliteId} / {beaconId} as a SatDbBeacon data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: The requested SatDbBeacon data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/beacons

Creates a new satellite beacon with the given attributes. {satelliteId} specifies the satellite for which the beacon shall be created. The unique *beaconId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbBeacon data object to store. Any *beaconId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbBeacon data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbBeacon data object added to the datbase. The replied data object contains the *beaconId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbBeacon data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Updates an existing satellite beacon with the given *satelliteId* / *beaconId* from the attached SatDbBeacon data object.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.
beaconId	Id of the beacon.

Expected Payload: The SatDbBeacon data object containing the fields to be updated in the database. If the data object contains values for *beaconId* and / or *satelliteId*, these values are ignored. *beaconId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbBeacon data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/beacons/{beaconId}

Remove a satellite beacon from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the beacon belongs to.
beaconId	Id of the beacon to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the beacon has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite beacon exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/beacons

Remove all satellite beacons which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all beacons for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tcchans

Returns the list of satellite TC channels for a given satellite as an array of SatDbTc data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite TC channels as an array ofSatDbTc data objects. This array may be empty if there are no TC channels defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Returns the information stored for a particular satellite TC channel addressed by {satelliteId} / {tcId} as a SatDbTc data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
tcId	Id of the TC channel.

Expected Payload: none

Replied Payload: The requested SatDbTc data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite channel.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite TC channel exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tcchans

Creates a new satellite TC channel with the given attributes. {satelliteId} specifies the satellite for which the TC channel shall be created. The unique *tcId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTc data object to store. Any *tcId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satelliteId*, the newly created SatDbTc data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTc data object added to the datbase. The replied data object contains the *tcId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTc data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Updates an existing satellite TC channel with the given *satelliteId* / *tcId* from the attached SatDbTc data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
tcId	Id of the TC channel.

Expected Payload: The SatDbTc data object containing the fields to be updated in the database. If the data object contains values for *tcId* and / or *satelliteId*, these values are ignored. *tcId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTc data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite satellite TC channel exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tcchans/{tcId}

Remove a satellite TC channel from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TC channel belongs to.
tcId	Id of the TC channel to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the TC channel has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite TC channel exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tcchans

Remove all satellite TC channels which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TC channels for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tmchans

Returns the list of satellite TM channels for a given satellite as an array of SatDbTm data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite TM channels as an array ofSatDbTm data objects. This array may be empty if there are no tm channels defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Returns the information stored for a particular satellite TM channel addressed by {satelliteId} / {tmId} as a SatDbTm data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
tmId	Id of the TM channel.

Expected Payload: none

Replied Payload: The requested SatDbTm data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite channel.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tmchans

Creates a new satellite TM channel with the given attributes. {satelliteId} specifies the satellite for which the tm channel shall be created. The unique *tmId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTm data object to store. Any *tmId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satelliteId*, the newly created SatDbTm data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTm data object added to the datbase. The replied data object contains the *tmId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTm data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Updates an existing satellite TM channel with the given *satelliteId* / *tmId* from the attached SatDbTm data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
tmId	Id of the TM channel.

Expected Payload: The SatDbTm data object containing the fields to be updated in the database. If the data object contains values for *tmId* and / or *satelliteId*, these values are ignored. *tmId* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTm data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tmchans/{tmId}

Remove a satellite TM channel from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TM channel belongs to.
tmId	Id of the TM channel to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the TM channel has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite satellite TM channel exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tmchans

Remove all satellite TM channels which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TM channels for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/i11params

Returns the list of I11 parameter sets for a given satellite as an array of SatDbI11Data data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of I11 parameter sets as an array ofSatDbI11Data data objects. This array may be empty if there are no I11 parameter sets defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Returns the information stored for a particular I11 parameter set addressed by {satelliteId} / {i11Id} as a SatDbI11Data data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
i11Id	Id of the I11 parameter set.

Expected Payload: none

Replied Payload: The requested SatDbI11Data data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case anApiError document is returned, describing details of the error.

Code	Description
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/i11params

Creates a new I11 parameter set with the given attributes. {satelliteId} specifies the satellite for which the I11 parameter set shall be created. The unique *i11Id* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbI11Data data object to store. Any *i11Id* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbI11Data data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbI11Data data object added to the datbase. The replied data object contains the *i11Id* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbI11Data data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Updates an existing I11 parameter set with the given *satelliteId* / *i11Id* from the attached SatDbI11Data data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Parameter	Description
i11Id	Id of the I11 parameter set.

Expected Payload: The SatDbI11Data data object containing the fields to be updated in the database. If the data object contains values for *i11Id* and / or *satelliteId*, these values are ignored. *i11Id* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbI11Data data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/i11params/{i11Id}

Remove a I11 parameter set from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the I11 parameter set belongs to.
i11Id	Id of the I11 parameter set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description

Code	Description
204	NO CONTENT, the I11 parameter set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite I11 parameter set exists with the given *satelliteId* / *i11Id*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/i11params

Remove all I11 parameter sets which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all I11 parameter sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satellites/{satelliteId}/tleparams

Returns the list of TLE parameter sets for a given satellite as an array of SatDbTLEData data objects.

Parameters:

Parameter	Description

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of TLE parameter sets as an array ofSatDbTLEData data objects. This array may be empty if there are no TLE parameter sets defined for this satellite. As there may be actually only one TLE parameter set for each satellite, the array may contain only zero or one elements.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Returns the information stored for a particular TLE parameter set addressed by {satelliteId} / {noradNumber} as a SatDbTLEData data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
noradNumber	Id / norad number of the TLE parameter set.

Expected Payload: none

Replied Payload: The requested SatDbTLEData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satellites/{satelliteId}/tleparams

Creates a new TLE parameter set with the given attributes. {satelliteId} specifies the satellite for which the TLE parameter set shall be created. The unique *noradNumber* is taken from the *noradNumber* field in the satellite's master data and returned in the response document. Hence it is important that the *noradNumber* field in the satellite's master data has been set before to the correct value.

As there may be only TLE parameter set assigned to a satellite, this POST method behaves in a special way: If there already exists a TLE parameter set for this satellite, it will be silently overwritten by the POST call. The call proceeds, stores the new data and returns a 201 return code. This behavior ensures, that not more than one TLE parameter set can be assigned to a satellite.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: The SatDbTLEData data object to store. Any *noradNumber* definition in the source is ignored as this will be taken from the *noradNumber* field in the satellite's master data when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbTLEData data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTLEData data object added to the datbase. The replied data object contains the *noradNumber* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTLEData data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Updates an existing TLE parameter set with the given *satelliteId* / *noradNumber* from the attached SatDbTLEData data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.
noradNumber	Id of the TLE parameter set.

Expected Payload: The SatDbTLEData data object containing the fields to be updated in the database. If the data object contains values for *noradNumber* and / or *satelliteId*, these values are ignored. *noradNumber* and *satelliteId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTLEData data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tleparams/{noradNumber}

Remove a TLE parameter set from the database.

Parameters:

Parameter	Description
satelliteId	Id of the satellite the TLE parameter set belongs to.
noradNumber	Id of the TLE parameter set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the TLE parameter set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite TLE parameter set exists with the given *satelliteId* / *noradNumber*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/tleparams

Remove all TLE parameter sets which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all TLE parameter sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satellites/{satelliteId}/ttracks

In general, the table track ('ttrack') API calls are located at /api/v1/antennas. The only call located at /api/v1/satellites is to temove all table track data sets which belong to the given *satelliteId* (for all antennas).

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

8.2.27 /api/v1/satoperators

This API endpoint gives access to various functions of the sat-nms satoperator database table, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/satoperators

Returns the list of stored satoperators as an array of SatDbSatOperator data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored satoperators as an array of SatDbSatOperator data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all satoperators and their attributes.

Code	Description
401	not logged in. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/satoperators/{operatorId}

Returns the basic properties of the satoperator addressed by {operatorId} as a SatDbSatOperator data object.

Parameters:

Parameter	Description
operatorId	Id of the satoperator.

Expected Payload: none

Replied Payload: The requested SatDbSatOperator data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satoperator.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satoperator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/satoperators

Creates a new satoperator with the given attributes. The unique *operatorId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbSatOperator data object to store. Any *operatorId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbSatOperator data object added to the datbase. The replied data object contains the *operatorId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbSatOperator data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/satoperators/{operatorId}

Updates an existing satellite operator with the given *operatorId* from the attached SatDbSatOperator data object.

Parameters:

Parameter	Description
operatorId	Id of the satoperator.

Expected Payload: The SatDbSatOperator data object containing the fields to be updated in the database. If the data object contains an *operatorId* key, the value is ignored. The database record is solely identified by *operatorId* set in the URL, the *operatorId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbSatOperator data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the satellite operator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/satoperators/{operatorId}

Remove a satoperator from the database. This deletes the satoperator's record from the satoperator table and removes all references to the deleted satoperator in the *satellites* table.

Parameters:

Parameter	Description
operatorId	Id of the satoperator to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the satellite operator with the given *operatorId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

8.2.28 /api/v1/channels

This API endpoint gives access to various functions of the*channels* database table, permitting to read, add, change or delete entries at the database. This database table holds all satellite channels known in the satnms database.

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/channels/

Returns the list of all satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/ip

Returns the list of all 'IP' type satellite channel records as an array of ChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array of ChannelData data objects. This array may be empty if there are no such channel records in the database..

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/asi

Returns the list of all 'ASI' type satellite channel records as an array of ChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array of ChannelData data objects. This array may be empty if there are no such channel records in the database.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/sat

Returns the list of all 'SAT' type satellite channel records as an array ofChannelData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no such channel records in the database.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/sat/{satelliteId}

Returns the list of 'SAT' type satellite channel records for a given satellite as an array of ChannelData data objects.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of satellite channel records as an array ofChannelData data objects. This array may be empty if there are no channel records defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/channels/{channelId}

Returns the information stored for a particular channel record addressed by {channelId} as a ChannelData data object.

Parameters:

Parameter	Description
channelId	Id of the channel record.

Expected Payload: none

Replied Payload: The requested ChannelData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/channels

Creates a new channel record with the given attributes.

Parameters: none

Expected Payload: The ChannelData data object to store. The channel definition must conform the following conditions:

- The *decoderInput* field must be set to one of 'SAT', 'IP' or 'ASI' to define the type of channel.
- The *name* field must be set to a value which is unique for the given satellite or the API call will decline to create the database record.
- 'SAT' type records must contain a valid *satelliteId*
- Any *channelId* definition in the source is ignored as this will be set by the database when the new entry is created.

Replied Payload: The ChannelData data object added to the datbase. The replied data object contains the *channelId* assigned to the newly created record.

Return Codes:

Code	Description

Code	Description
201	CREATED, the created ChannelData data object is replied as payload
400	BAD_REQUEST if the supplied ChannelData does not conform the conditions listed above.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/channels/{channelId}

Updates an existing channel record with the given *channelId* from the attached ChannelData data object.

Parameters:

Parameter	Description
channelId	Id of the channel record.

Expected Payload: The ChannelData data object containing the fields to be updated in the database. Fields not contained in the supplied data object are retained unchanged. The channel definition must conform the following conditions:

- If it changes the *name* of the record, this must be set to a value which is unique for the given satellite or the API call will decline to create the database record.
- 'SAT' type records must contain a valid *satelliteId*
- Any *channelId* definition in the source is ignored as the *channelId* is immutable once a record has been created.

Replied Payload: The ChannelData data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
400	BAD_REQUEST if the supplied ChannelData does not conform the conditions listed above.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/channels/{channelId}

Remove a channel record from the database.

Parameters:

Parameter	Description
channelId	Id of the channel record to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the channel record has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite channel record exists with the given *channelId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/channels/sat/{satelliteId}

Remove all channel records which belong to the given *satelliteId*.

Parameters:

Parameter	Description
satelliteId	Id of the satellite

Return Codes:

Code	Description
204	NO CONTENT, all channel records for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

8.2.29 /api/v1/streamkeys

This API endpoint gives access to various functions of the *streamkeys* database table, permitting to read, add, change or delete entries at the database. This database table holds all BISS-1/BISS-E keys and SRT passphrases known in the sat-nms database.

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/streamkeys

Returns the list of stored stream keys as an array of StreamKeyData data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored stream keys as an array of StreamKeyData data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all streamkeys and their attributes.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/streamkeys/{keyId}

Returns the basic properties of the stream key addressed by {keyId} as aStreamKeyData data object.

Parameters:

Parameter	Description
keyId	Id of the stream key.

Expected Payload: none

Replied Payload: The requested StreamKeyData data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested stream key.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	stream key with the given *keyId* does not exists. In this case anApiError document is returned, describing details of the error.

POST /api/v1/streamkeys

Creates a new stream key with the given attributes. The unique*keyId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The StreamKeyData data object to store. Any *keyId* definition in the source is ignored as this will be set by the database when the new entry is created. The *name* field in the supplied payload must be unique over the entire table or the API call will decline to create the database record.

Replied Payload: The StreamKeyData data object added to the database. The replied data object contains the *keyId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created StreamKeyData data object is replied as payload
400	BAD_REQUEST. Either the *name* supplied with the request is not unique or the*key* does not conform to the format restrictions for the given *keyType*

Code	Description
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/streamkeys/{keyId}

Updates an existing stream key with the given *keyId* from the attached StreamKeyData data object.

Parameters:

Parameter	Description
keyId	Id of the stream key.

Expected Payload: The StreamKeyData data object containing the fields to be updated in the database. If the data object contains an *keyId* key, the value is ignored. The database record is solely identified by *keyId* set in the URL, the *keyId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged. The *name* field in the supplied payload must be unique over the entire table or the API call will decline to modify the database record.

Replied Payload: The StreamKeyData data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
400	BAD_REQUEST. Either the *name* supplied with the request is not unique or the*key* does not conform to the format restrictions for the given *keyType*
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the stream key with the given *keyId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/streamkeys/{keyId}

Remove a stream key from the database.

Parameters:

Parameter	Description
keyId	Id of the stream key to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the stream key with the given *keyId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

8.2.30 /api/v1/antennas

This API endpoint gives access to various functions of the sat-nms antenna database table, permitting to read, add, change or delete entries at the database

Supported HTTP methods: GET, POST, PATCH, DELETE, HEAD, OPTIONS

The API generally does not permit PATCH or DELETE operations on entire tables, this to protect the database from being accidentally be deleted. Even POST operations only work with single data objects.

GET /api/v1/antennas

Returns the list of stored antennas as an array of SatDbAntenna data objects.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored antennas as an array of SatDbAntenna data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document with all antennas and their attributes.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}

Returns the basic properties of the antenna addressed by {antennaId} as a SatDbAntenna data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The requested SatDbAntenna data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested antenna.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/antennas

Creates a new antenna with the given attributes. The unique *antennaId* is generated by the database backend and returned in the response document.

Parameters: none

Expected Payload: The SatDbAntenna data object to store. Any *antennaId* definition in the source is ignored as this will be set by the database when the new entry is created. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbAntenna data object added to the database. The replied data object contains the *antennaId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbAntenna data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}

Updates an existing antenna with the given *antennaId* from the attached SatDbAntenna data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: The SatDbAntenna data object containing the fields to be updated in the database. If the data object contains an *antennaId* key, the value is ignored. The database record is solely identified by *antennaId* set in the URL, the *antennaId* of the record is immutable. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbAntenna data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}

Remove a antenna from the database. This deletes the antenna's record from the antenna table and all records from the *satpositions* table which refer to the deleted antenna.

Parameters:

Parameter	Description
antennaId	Id of the antenna to delete

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite has been deleted.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/positions

Returns the list of satellite positions for a given antenna as an array of SatDbPosition data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The list of satellite positions as an array of SatDbPosition data objects. This array may be empty if there are no satellite positions defined for this antenna.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	the antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/positions/{positionId}

Returns the information stored for a particular satellite position addressed by {antennaId} / {positionId} as a SatDbPosition data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
positionId	Id of the satellite position.

Expected Payload: none

Replied Payload: The requested SatDbPosition data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested antenna.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.

ST /api/v1/antennas/{antennaId}/positions

Creates a new satellite position with the given attributes. {antennaId} specifies the antenna for which the satellite position shall be created. The unique *positionId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: The SatDbPosition data object to store. Any *positionId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *antennaId*, the newly created SatDbPosition data object will be linked to the antenna specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbPosition data object added to the database. The replied data object contains the *positionId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbPosition data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/positions/{positionId}

Updates an existing satellite position with the given *antennaId* / *positionId* from the attached SatDbPosition data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
positionId	Id of the satellite position.

Expected Payload: The SatDbPosition data object containing the fields to be updated in the database. If the data object contains values for *positionId* and / or *antennaId*, these values are ignored. *positionId* and *antennaId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbPosition data object read from the database after the requested changes have been applied.

Return Codes:

Code	Description

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/positions/{positionId}

Remove a satellite position from the database.

Parameters:

Parameter	Description
antennaId	Id of the antenna the satellite position belongs to.
positionId	Id of the satellite position to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the satellite position has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no antenna satellite position exists with the given *antennaId* / *positionId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/positions

Remove all satellite positions which belong to the given *antennaId*.

Parameters:

Parameter	Description
antennaId	Id of the antenna

Return Codes:

Code	Description
204	NO CONTENT, all satellite positions for the selected antenna have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	antenna with the given *antennaId* does not exist. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/ttracks

Returns the list of table track data sets for a given antenna as an array of SatDbTableTracking data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.

Expected Payload: none

Replied Payload: The list of table track data sets as an array of SatDbTableTracking data objects. This array may be empty if there are no table track data sets defined for this antenna.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	antenna with the given *antennaId* does not exists. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Returns the list of table track data sets for a given satellite/antenna combination as an array of SatDbTableTracking data objects.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The list of table track data sets as an array ofSatDbTableTracking data objects. This array may be empty if there are no table track data sets defined for this satellite.

Return Codes:

Code	Description
200	OK, returns a JSON document with the requested information
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

GET /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Returns the information stored for a particular table track data set addressed by {antennaId} / {satelliteId} / {ttId} as a SatDbTableTracking data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.
ttId	Id of the table track data set.

Expected Payload: none

Replied Payload: The requested SatDbTableTracking data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite table track data set exists with the given *satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Creates a new table track data set with the given attributes. {antennaId} / {satelliteId} specifies the satellite / antenna combination for which the table track data set shall be created. The unique *ttId* is generated by the database backend and returned in the response document.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.

Expected Payload: The SatDbTableTracking data object to store. Any *ttId* definition in the source is ignored as this will be set by the database when the new entry is created. The same applies to *satellite_Id*, the newly created SatDbTableTracking data object will be linked to the satellite specified in the URL. If other keys of the data object are missing, default values will be assigned these keys.

Replied Payload: The SatDbTableTracking data object added to the datbase. The replied data object contains the *ttId* assigned to the newly created record.

Return Codes:

Code	Description
201	CREATED, the created SatDbTableTracking data object is replied as payload
401	not logged in. In this case an ApiError document is returned,describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Updates an existing table tracking data set with the given *antennaId* / *satelliteId* / *ttId* from the attached SatDbTableTracking data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite.
ttId	Id of the table track data set.

Expected Payload: The SatDbTableTracking data object containing the fields to be updated in the database. If the data object contains values for any of *ttId*, *satelliteId* or *antennaId*, these values are ignored. *ttId*, *satelliteId* and *antennaId* are immutable once a record has been created. Keys not contained in the supplied data object are retained unchanged.

Replied Payload: The SatDbTableTracking data object read from the datbase after the requested changes have been applied.

Return Codes:

Code	Description
200	OK, changes have been applied.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no satellite table track data set exists with the given *satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/{satelliteId}/ttracks/{ttId}

Remove a table track data set from the database.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
satelliteId	Id of the satellite the table track data set belongs to.

Parameter	Description
ttId	Id of the table track data set to delete.

Expected Payload: none

Replied Payload: none

Return Codes:

Code	Description
204	NO CONTENT, the table track data set has been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	no satellite table track data set exists with the given *satelliteId* / *ttId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/{satelliteId}/ttracks

Remove all table track data sets which belong to the given *satelliteId* / *antennaId* combination.

Parameters:

Parameter	Description
satelliteId	Id of the satellite
antennaId	Id of the antenna.

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case an ApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/ttracks

Remove all table track data sets which belong to the given antenna.

Parameters:

Parameter	Description
antennaId	Id of the antenna

Return Codes:

Code	Description
204	NO CONTENT, all table track data sets for the selected satellite have been deleted.
401	not logged in. In this case anApiError document is returned,describing details of the error.
404	satellite with the given *satelliteId* does not exist. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Returns the attenuation value addressed by {antennaId} / {beaconId} as a SatDbBeaconAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: The requested SatDbBeaconAttenuation data object. If no attenuation has

been stored for this antennaId/beaconId combination, an empty SatDbBeaconAttenuation data object with the attenuation value set to 0.0 is returned.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Sets the attenuation value for an {antennaId} / {beaconId} combination. Returns the attenuation set as a SatDbBeaconAttenuation data object. If no attenuation record exists for this antenna / beacon combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: The attenuation to set as a SatDbBeaconAttenuation data object. Only the attenuation is read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbBeaconAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Sets the attenuation value for an {antennaId} / {beaconId} combination. Returns the attenuation set as a SatDbBeaconAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: The attenuation to set as a SatDbBeaconAttenuation data object. Only the attenuation is read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbBeaconAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/beacon-atten/{beaconId}

Deletes the attenuation value addressed by {antennaId} / {beaconId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
beaconId	Id of the beacon.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or beaconId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *beaconId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Returns the attenuation values addressed by {antennaId} / {tcId} as a SatDbTcAttenuation data object. If no attenuation has been stored for this antennaId/tcId combination, an empty SatDbTcAttenuation data object with the attenuation values set to 0.0 is returned.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: none

Replied Payload: The requested SatDbTcAttenuation data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Sets the attenuation values for an {antennaId} / {tcId} combination. Returns the attenuation set as a SatDbTcAttenuation data object. If no attenuation record exists for this antenna / tc combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: The attenuation to set as a SatDbTcAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTcAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Modifies the attenuation values for an {antennaId} / {tcId} combination. Returns the attenuation set as a SatDbTcAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: One or both attenuation values to set in aSatDbTcAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTcAttenuation data object with the updated attenuation.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/tc-atten/{tcId}

Deletes the attenuation values addressed by {antennaId} / {tcId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tcId	Id of the tc channel.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or tcId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tcId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Returns the attenuation values addressed by {antennaId} / {tmId} as a SatDbTmAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: none

Replied Payload: The requested SatDbTmAttenuation data object. If no attenuation has been stored for this antennaId/tcId combination, an empty SatDbTmAttenuation data object with the attenuation values set to 0.0 is returned.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.

Code	Description
401	not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Sets the attenuation values for an {antennaId} / {tmId} combination. Returns the attenuation values set as a SatDbTmAttenuation data object. If no attenuation record exists for this antenna / tm combination a new one is created. If there is already an attenuation record for this combination, the existing one will be overwritten.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: The attenuation to set as a SatDbTmAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTmAttenuation data object with the updated attenuation .

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Modifies the attenuation values for an {antennaId} / {tmId} combination. Returns the attenuation set as a SatDbTmAttenuation data object.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: One or both attenuation values to set in aSatDbTmAttenuation data object. Only the attenuation values are read from this data object, any IDs passed to this API function are ignored. The IDs are defined solely by the URL parameters.

Replied Payload: The SatDbTmAttenuation data object with the updated attenuation.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/antennas/{antennaId}/tm-atten/{tmId}

Deletes the attenuation values addressed by {antennaId} / {tmId}.

Parameters:

Parameter	Description
antennaId	Id of the antenna.
tmId	Id of the tm channel.

Expected Payload: none

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the record has been deleted
400	BAD_REQUEST, antennaId or tmId could not be parsed to a numeric identifier. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.
404	no attenuation record exists with the given *satelliteId* / *tmId*. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

8.2.31 /api/v1/satdetails

This API endpoint reports all relevant data for a satellite in a compact form. It has been added for convenience, unburdening the front end programmer from the need to gather this information from different database tables using multiple API calls.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/satdetails/{satelliteId}

Returns detailed properties of the satellite addressed by {satelliteId} as a SatDbSatDetails data object.

Parameters:

Parameter	Description
satelliteId	Id of the satellite.

Expected Payload: none

Replied Payload: The requested SatDbSatDetails data object.

Return Codes:

Code	Description
200	OK, returns a document describing the requested satellite.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	satellite with the given *satelliteId* does not exists. In this case anApiError document is returned, describing details of the error.

8.2.32 /api/v1/tleimport

This API endpoint lets you upload and import a list of TLE parameter sets from a data file. The expected file format is compatible to the well known 'geo.txt' file from celestrak.com. The file must contain three lines of text for each satellite, the firs line contains the satellite name, followed by two lines in TLE format. The norad number contained in both lines of the TLE format is used to identify for which satellite each record shall be used. The satellite name is treated as a comment only.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/tleimport

Expects a SatDbTleImport document containing the file to import along with some additional parameters.

Parameters: none

Expected Payload: The SatDbTleImport document containing the file to import and the import mode to use.

Replied Payload: none, except for errors.

Return Codes:

Code	Description
204	OK, the file has been imported to the database. Do data is returned.
400	The file could not be parsed. In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

8.2.33 /api/v1/eventreport

This API endpoint lets you create reports from the M&C system event database. The M&C systems managed by the backend share a common event database which store all faults, warnings and informational messages occurring in the particular M&C systems.

A report from the event database is created by sending anEventReportRequest document with

a POST call to this API endpoint. The EventReportRequest defines the time span to read and some filter settings which allow to read a set of messages with a specific content.

The API call responds with a EventReport document containing the requested event report.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/eventreport

Expects an EventReportRequest document containing the time span to read and the report filter settings.

Parameters: none

Expected Payload: The EventReportRequest document containing the time span to read and the report filter settings.

Replied Payload: The EventReport document containing the requested report.

Return Codes:

Code	Description
200	OK, the request has been processed, the report generated is replied.
500	The event database does not reply (internal error). In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.34 /api/v1/eventack

The sat-nms event database provides a "not yet acknowledged" flag with each event in the database. This flag is initially set for all faults. This API endpoint gives access to the "acknowledge" function which clears this flag for a single event or for all events which have the flag set.

To acknowledge on or all events, the front end has to send anEventAck document with a POST call to this API end point. The EventAck document defines if all events or a particular one shall be acknowledged.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/eventack

Expects an EventAck document which defines if all events or a particular one shall be acknowledged.

Parameters: none

Expected Payload: The EventAck document which defines if all events or a particular one shall be acknowledged.

Replied Payload: none, except for errors.

Return Codes:

Code	Description
204	OK, the request has been processed. Do data is returned. There is no error checking, acknowledging a non-existing event or an event which does not require acknowledging is not treated as a fault.
500	The event database does not reply (internal error). In this case anApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.35 /api/v1/trackmem

When subscribing for a '.state.mode4' parameter of a SatService ACU-ODM, the front end receives a message which indicates 'a new tracking history for this antenna is available' instead of sending the bulky tracking history itself over the websocket connection.

After the front end received this message, it can use the /api/v1/trackmem API call to retrieve the full tracking history as a TrackingHistory data object.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/trackmem/{odmName}

Parameters:

Parameter	Description
odmName	The device name of the ODM for which the tracking history shall be read. This must be a fully qualified name including name of the M&C where the ODM is managed.

Expected Payload: none

Replied Payload: A TrackingHistory document containing the tracking history of the referenced antenna controller.

Return Codes:

Code	Description

Code	Description
200	OK, the requested TrackingHistory document is returned.
400	Either the requested device does not exist or the device has not yet sent a tracking history. This may happen if the front end did no subscribe for the 'state.mode4' of the ODM before or if the ODM is not operational. In this case an ApiError document is returned, describing details of the error.
401	not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.36 /api/v1/treeview

This API end point lets you retrieve the treeview data structure as a whole or partially by requesting only a branch of the tree. The API also provides a PUT method for this end point which lets you replace the user defined part of the tree for a given M&C name.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/treeview

Gets the complete tree definition for all M&C systems configured in the backend. Includes the auto-generated and the user defined parts of the tree definition

Parameters: none

Expected Payload: none

Replied Payload: A TreeViewNode document containing the complete tree view tree, starting with the ROOT node of the tree.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/treeview/{mncName}

Gets the tree definition for a given M&C system. Includes the user defined part of the tree definition as well as the device list for this M&C system.

Parameters:

Parameter	Description

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be read.

Expected Payload: none

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C, starting with the MNCNODE node for this M&C.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
404	The requested M&C system does not exist or is actually offline. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/treeview/{mncName}/{subsystemName}

Gets the tree definition for a given M&C subsystem.

Parameters:

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be read.
subsystemName	The name of the subsystem to retrieve. This subsystem name also may address nested subsystems, e.g. `SUBSYS-1.SUB_SYS-2` denotes a subsystem `SUSBSYS-2` contained in `SUBSYS-1`

Expected Payload: none

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C subsystem, starting with the SUBSYSTEM node for this M&C subsystem.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.

Code	Description
404	Either the requested M&C system does not exist or is actually offline or no subsystem with the given name exists in this M&C system. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/treeview/{mncName}

Replaces the tree definition for a given M&C system with a new version supplied by the client. Echoes the tree definition for this M&C system after the request has been processed.

Parameters:

Parameter	Description
mncName	The name of the M&C for which the tree definition shall be replaced.

Expected Payload: A TreeViewNode document containing the user defined part tree branch for the given M&C, starting with the MNCNODE node for this M&C. Only the SUBSYSTEM children of this node are processed, if the TReeViewNode has a DEVICELIST child, this is ignored.

Replied Payload: A TreeViewNode document containing the tree branch for the given M&C, starting with the MNCNODE node for this M&C. This is the tree view branch after the request has been processed, it includes the DEVICELIST for this M&C.

Return Codes:

Code	Description
200	OK, the requested TreeViewNode document is returned.
404	The requested M&C system does not exist or is actually offline. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.37 /api/v1/debug

This API end point lets you send debug commands to the (already opened) debug terminal of a M&C.

Supported HTTP methods: POST, HEAD, OPTIONS

POST /api/v1/debug/{mncName}

Sends one line containing a debug command to the debug terminal of the given M&C. The websocket connection for the debug terminal must already be open to make the command work. Echoes the original command after the request has been processed.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the debug command to.

Expected Payload: A DebugMessage document containing the line with the command to send.

Replied Payload: A copy of the DebugMessage which has been sent.

Return Codes:

Code	Description
200	OK, the echoed DebugMessage document is returned.
404	The requested M&C system does not exist, is actually offline or another user is already connected to the debug terminal at this M&C. In this case an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.38 /api/v1/redundancy

This API end point lets you control or interrogate the redundancy properties for redundant M&Cs.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/redundancy/{mncName}

Queries the state of the redundancy control logic for the given M&C

Parameters:

Parameter	Description
mncName	The name of the M&C to query.

Expected Payload: none.

Replied Payload: A RedundancyState object reporting the redundancy properties of the M&C.

Return Codes:

Code	Description
200	OK, a RedundancyState document is returned.
400	The requested M&C system is not configured as a redundant pair. In this case an ApiError document is returned, describing details of the error.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/redundancy

Queries the state of the redundancy control logic for all redundant M&Cs.

Parameters: none.

Expected Payload: none.

Replied Payload: A RedundancySummary object reporting the redundancy properties of all redundant M&Cs.

Return Codes:

Code	Description
200	OK, a RedundancySummary document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/redundancy/{mncName}

Sends a redundancy command to either switch a redundant M&C pair or to enable/disable the M&C redundancy.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the redundancy command to.

Expected Payload: A RedundancyCmd document containing the command to send.

Replied Payload: A copy of the RedundancyCmd which has been sent.

Return Codes:

Code	Description
200	OK, the echoed RedundancyCmd document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

8.2.39 /api/v1/restart

This API end point lets you send a restart command to a given M&C.

Supported HTTP methods: PUT, HEAD, OPTIONS

PUT /api/v1/restart/{mncName}

Sends a restart command to the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C to send the redundancy command to.

Expected Payload: A RestartCmd document with *cmd* set to 'RESTART'. With other values of *cmd* the restart command is ignored.

Replied Payload: A copy of the RestartCmd which has been sent.

Return Codes:

Code	Description
200	OK, the echoed RestartCmd document is returned.
400	Either the RestartCmd passed to the call did not contain cmd=RESTART or the requested M&C system is offline and cannot be restarted. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.40 /api/v1/devicesetup

This API end point lets read and write the device setup for a specific M&C.

Writing a device setup as a whole to the M&C requires a M&C restart to make the M&C read and interpret the changed setup. Hence this is to be considered a legacy operation which will be replaced by API calls to modify the device configuration of the M&C 'on the fly' without restarting it, as soon as these API calls are available.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/devicesetup/{mncName}

Get the device setup for the given M&C.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the device setup from.

Expected Payload: none.

Replied Payload: A DeviceSetup document describing the device configuration of the given M&C.

Return Codes:

Code	Description
200	OK, a DeviceSetup document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/devicesetup/{mncName}

Sends a new device configuration to the given M&C. The configuration will be saved as

'vlc.setup' on this M&C, replacing the existing configuration. The M&C will use this configuration after the next restart (see /api/v1/restart).

Parameters:

Parameter	Description
mncName	The name of the M&C to send the device setup to.

Expected Payload: The DeviceSetup document containing the new / modified device configuration.

Replied Payload: Echoes back the DeviceSetup which has been sent.

Return Codes:

Code	Description
200	OK, the echoed DeviceSetup document is returned.
404	The requested M&C system does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.41 /api/v1/drivers

This API end point returns a list of the available device drivers for a specific M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/drivers

Get the list of the available device drivers. The list is fetched from the primary M&C.

Parameters: none

Expected Payload: none.

Replied Payload: An ItemList document containing the names of all device drivers available.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.

Code	Description
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.42 /api/v1/protocols

This API end point returns a list of the available device communication protocols. The list is fetched from the primary M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/protocols

Get the list of all available device communication protocols. The list is fetched from the primary M&C.

Parameters: none.

Expected Payload: none.

Replied Payload: An ItemList document containing the names of all device communication protocols available.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.43 /api/v1/preferredprotocols

This API end point returns a list of the preferred device communication protocols a particular device driver uses. The list is fetched from the primary M&C.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/preferredprotocols/{driverName}

Get the list of the preferred device communication protocols a particular device driver uses.

Actually a ***sat-nms*** devices driver defines exactly one preferred protocol, hence this list will always have exactly one element. Future ***sat-nms*** versions may permit to define multiple preferred protocols for a driver.

Parameters:

Parameter	Description
driverName	The name of the device driver to check.

Expected Payload: none.

Replied Payload: An ItemList document containing the names of the preferred device communication protocols of the given driver.

Return Codes:

Code	Description
200	OK, an ItemList document is returned.
400	The primary M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
404	The device driver this call refers to does not exist. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

8.2.44 /api/v1/filerecordings

This API end point returns the recorded data for a File-Recorder device

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/filerecordings/{mncName}/{deviceName}

Reads the recorded data from the given File-Recorder device

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the File-Recorder device to read the data from

Expected Payload: none.

Replied Payload: The recorded data in CSV format (text/plain). Columns are separated by comma characters, the first column contains the time stamp in 'YYYY-MM-DD HH:MM:SS' format (UTC). All other columns contain the recorded data. Example:

```
2023-02-16 09:12:38,28.3,,,
2023-02-16 09:13:39,28.4,,,
2023-02-16 09:14:39,28.5,,,
2023-02-16 09:15:38,28.4,,,
2023-02-16 09:16:38,28.5,,,
2023-02-16 09:17:38,28.3,,,
2023-02-16 09:18:38,28.4,,,
2023-02-16 09:19:38,28.4,,,
```

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

8.2.45 /api/v1/filerecordersettings

Reads and stores the diagram settings for a given file recorder device.

Supported HTTP methods: GET, PUT, HEAD, OPTIONS

GET /api/v1/filerecordersettings/{mncName}/{deviceName}

Reads the diagram settings for a given file recorder device.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the File-Recorder device to read the data from

Expected Payload: none.

Replied Payload: The diagram settings as a FRViewProperties document.

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

PUT /api/v1/filerecordersettings/{mncName}/{deviceName}

Stores the diagram settings for a given file recorder device.

Parameters:

Parameter	Description
mncName	The name of the M&C to store the data to
deviceName	The name of the File-Recorder device to store the data for

Expected Payload: The diagram settings to store as a FRViewProperties document.

Replied Payload: Echoes back the stored data (after storing it at the destination M&C and reading it back from there).

Return Codes:

Code	Description
200	OK, the echoed back diagram settings document is returned.
400	The M&C is offline, data cannot be stored. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	Either the referred M&C or the device does not exists or the device is not a File-Recorder device. In this case an ApiError document is returned, describing details of the error.

8.2.46 /api/v1/acutargets

Reads the targets / satellites stored on a given ACU.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/acutargets/{mncName}/{deviceName}

Reads the targets / satellites stored on a given ACU.

Parameters:

Parameter	Description
mncName	The name of the M&C to read the data from
deviceName	The name of the device to read the data from. This may be either the name of the ACU device itself or the name of the Antenna-Management device controlling the ACU

Expected Payload: none.

Replied Payload: The target list of the queries ACU as a AcuTargetList document.

Return Codes:

Code	Description
200	OK, the requested file recorder data is returned.
400	The M&C is offline, cannot be queried. In this case anApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Either the referred M&C or the device does not exists or the device does not provide a target list. In this case an ApiError document is returned, describing details of the error.

8.2.47 /api/v1/inventory

This API endpoint is used to inspect or modify the backend's inventory of items / devices.

Supported HTTP methods: GET, POST, PATCH, HEAD, OPTIONS

GET /api/v1/inventory/{itemId}

Gets the inventory item with the given ID.

Parameters:

Parameter	Description
itemId	The itemId of the item to get

Expected Payload: none.

Replied Payload: An InventoryItem record with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/inventory/{itemId}/devices

Gets the list of InventoryDevice objects mapped to the given inventory item.

Parameters:

Parameter	Description
itemId	The itemId of the item to get

Expected Payload: none.

Replied Payload: An array of InventoryDevice records mapped to the item with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryDevice Array is returned.

Code	Description
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

GET /api/v1/inventory/{itemId}/devices/{deviceId}

Gets the InventoryDevice objejct with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to get
deviceId	The deviceId of the device to get

Expected Payload: none.

Replied Payload: An InventoryDevice record with the given ID combination.

Return Codes:

Code	Description
200	OK, the requested InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/inventory

Creates a new InventoryItem entry in the database.

Parameters: none.

Expected Payload:An InventoryItem record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'itemId' is ignored. The POST command creates a new record, the database

assigns a unique itemId to the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created InventoryItem record

Return Codes:

Code	Description
200	OK, the modified InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

POST /api/v1/inventory/{itemId}/devices

Creates a new InventoryDevice entry in the database and attaches this to the given item.

Parameters:

Parameter	Description
itemId	The itemId of the item to add the new device to

Expected Payload:An InventoryDevice record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'deviceId' is ignored. The POST command creates a new record, the database assigns a unique deviceId to the created record.
- The field 'itemId' is ignored. The POST command always takes the the 'itemId' from the URL and sets this in the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created InventoryItem record

Return Codes:

Code	Description

Code	Description
200	OK, the modified InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/inventory/{itemId}

Modifies the inventory item with the given ID.

Parameters:

Parameter	Description
itemId	The itemId of the item to modify

Expected Payload:An InventoryItem record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'itemId' is ignored. The PATCH command modifies the record specified in the URI, the 'itemId' ofthis record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The InventoryItem record after modification

Return Codes:

Code	Description
200	OK, the modified InventoryItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

Code	Description
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/inventory/{itemId}/devices/{deviceId}

Modifies the InventoryDevice object with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to modify
deviceId	The deviceId of the device to modify

Expected Payload:An InventoryDevice record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'deviceId' is ignored. The PATCH command modifies the record specified in the URI, the 'deviceId' of this record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The InventoryDevice record after modification

Return Codes:

Code	Description
200	OK, the modified InventoryDevice document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/inventory/{itemId}

Deletes the inventory item with the given ID. As a side effect, allInventoryDevice and InventoryLogEntry objects referring to this inventory item are deleted as well.

Parameters:

Parameter	Description
itemId	The itemId of the item to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed InventoryItem has been deleted.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/inventory/{itemId}/devices/{deviceId}

Deletes the InventoryDevice obkect with the given ID combination.

Parameters:

Parameter	Description
itemId	The itemId of the item to delete
deviceId	The deviceId of the device to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed InventoryDevice has been deleted.

Code	Description
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

POST /api/v1/inventory/list

Gets a list of inventory items matching the filter information sent with the request.

Parameters: none

Expected Payload: An InventoryItem document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the item in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be modified at this time or later'.

Replied Payload: An array of InventoryItem records fulfilling the filter spec. The list is sorted by modification time, oldest first. If no records match the filter, an empty list is returned.

Return Codes:

Code	Description
200	OK, the requested InventoryItem array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	The backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/inventory/log/{entryId}

Gets the inventory log entry with the given ID.

Parameters:

Parameter	Description

Parameter	Description
entryId	The entryId of the log entry to get

Expected Payload: none.

Replied Payload: An InventoryLogEntry record with the given ID

Return Codes:

Code	Description
200	OK, the requested InventoryLogEntry document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No InventoryLogEntry with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/inventory/log

Adds a new InventoryLogEntry to the database.

Parameters: none.

Expected Payload:An InventoryLogEntry record containing the fields of the record to create. The following rules apply:

- log entries cannot be modified / rewritten. All fields must be supplied with their desired values.
- The field 'entryId' is ignored. The POST command creates a new record, the database assigns a unique itemId to the created record.
- The 'created' field is set to the time when the command is executed, a 'created' time specified in the supplied data is ignored.

Replied Payload: The created InventoryLogEntry record

Return Codes:

Code	Description
200	OK, the modified InventoryLogEntry document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	No InventoryLogEntry with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

POST /api/v1/inventory/log/list

Gets a list of inventory log entries matching the filter information sent with the request.

Parameters: none

Expected Payload: An InventoryLogEntry document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the entry in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be created at this time or later'.

Replied Payload: An array of InventoryLogEntry records fulfilling the filter spec. The list is sorted by creation time, oldest first.

Return Codes:

Code	Description
200	OK, the requested InventoryLogEntry array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	The backend does not support the inventory database. In this case anApiError document is returned, describing details of the error.

8.2.48 /api/v1/documents

This API endpoint is used to inspect or modify the backend's list of documents.

Supported HTTP methods: GET, POST, PATCH, HEAD, OPTIONS

GET /api/v1/documents/{documentId}

Gets the documents item with the given ID.

Parameters:

Parameter	Description

Parameter	Description
documentId	The documentId of the item to get

Expected Payload: none.

Replied Payload: An DocumentItem record with the given ID. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the requested DocumentItem document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/documents

Creates a new DocumentItem entry in the database.

Parameters: none.

Expected Payload:An DocumentItem record containing the fields of the record to create. The following rules apply:

- Fields which are null or missing are set to 'null' in the record.
- The field 'documentId' is ignored. The POST command creates a new record, the database assigns a unique documentId to the created record.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.

Replied Payload: The created DocumentItem record. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the modified DocumentItem document is returned.

Code	Description
400	The file already exists. In this case the file is not written and an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/documents/{documentId}

Modifies the documents item with the given ID.

Parameters:

Parameter	Description
documentId	The documentId of the item to modify

Expected Payload:An DocumentItem record containing the fields to modify. The following rules apply:

- Fields which are null or missing are not changed in the record.
- To set a field to 'null' set it to an empty string.
- The field 'documentId' is ignored. The PATCH command modifies the record specified in the URI, the 'documentId' of this record never changes.
- The 'modified' field is set to the time when the command is executed, a 'modified' time specified in the supplied data is ignored.
- If the field 'fileContent' is contained, the stored file gets rewritten with this content.
- If 'fileName' and 'fileContent' are changed at the same time, the old file is deleted before writing the new one.

Replied Payload: The DocumentItem record after modification. The DocumentItem does not contain its 'fileContent' field, use the '/api/v1/documents/content/' API call to retrieve the document content.

Return Codes:

Code	Description
200	OK, the modified DocumentItem document is returned.

Code	Description
400	A file with the new name already exists. In this case the file is not written and an ApiError document is returned, describing details of the error.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the documents database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

DELETE /api/v1/documents/{documentId}

Deletes the documents item with the given ID. This also deletes the file this record refers to.

Parameters:

Parameter	Description
documentId	The documentId of the item to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
204	OK, the addressed DocumentItem has been deleted.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.
500	DB is not writable. In this case an ApiError document is returned,describing details of the error.

GET /api/v1/documents/content/{documentId}

Gets the file content of the document item with the given ID.

Parameters:

Parameter	Description
documentId	The documentId of the item to get

Expected Payload: none.

Replied Payload: The file content of the data file this document record refers to.

Return Codes:

Code	Description
200	OK, the requested file is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No DocumentItem with this ID exists or the backend does not support the inventory database. In this case an ApiError document is returned, describing details of the error.

POST /api/v1/documents/list

Gets a list of documents items matching the filter information sent with the request.

Parameters: none

Expected Payload: An DocumentItem document containing the filter to be applied. Any fields in the document which are missing or null are intepreted as a wildcard. Specified fields must *all* match to include the item in the returned list. If a 'modified' timestamp is given in the filter description, it is interpreted as 'must be modified at this time or later'. The 'fileContent' field is ignored as a filter.

Replied Payload: An array of DocumentItem records fulfilling the filter spec. The list is sorted by file name, 'a' to 'z'. The DocumentItem records do not contain their 'fileContent' field, this API call is intended to list the items, not to to execute a bulk read of the documents.

Return Codes:

Code	Description
200	OK, the requested DocumentItem array is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.

Code	Description
404	The backend does not support the inventory database. In this case anApiError document is returned, describing details of the error.

8.2.49 /api/v1/thumbnail

This API call fetches a thumnail image from the backend. Thumbnail images are cached in the backend. If you subscribe for a thumbnail image via the websocket, the frontend only receives a messages which tells that a new thumbnail picture is available. The image itself then must be fetched with the `/api/v1/thumbnail` API call.

Supported HTTP methods: GET, HEAD, OPTIONS

GET /api/v1/thumbnail/{messageId}

Gets the thumbnail image for the given message ID (e.g. 'MNC-1.IRD-3.thumbnail') Please note, that the messageId must be subscribed or the API call will return a 404 Not Found because the backens has not subscribes for the thumbnail message at the M&C

Parameters:

Parameter	Description
messageId	The full message ID of the thumbnail parameter of the device

Expected Payload: none.

Replied Payload: An Thumbnail document containing the thumbnail image.

Return Codes:

Code	Description
200	OK, the requested Thumbnail document is returned.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	Not found. In this case an ApiError document is returned, describing details of the error.

8.2.50 /api/v1/backendinfo

This API endpoint gives the backend information for example name and role.

Supported HTTP methods: GET

GET /api/v1/backendinfo

Returns the BackendInfo data objects.

Parameters: none

Expected Payload: none

Replied Payload: BackendInfo data objects

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

8.2.51 /api/v1/globalSettings

This API endpoint is used to get and or save global settings for the users.

Supported HTTP methods: GET, PATCH, HEAD, OPTIONS

GET /api/v1/globalSettings

Gets the saved settings if not Backend should retrun default values with setting keys.

Parameters: none

Expected Payload: none

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case an ApiError document is returned,describing details of the error.

PATCH /api/v1/globalSettings

Save setting in the Database. If *setting key* already exists it will overwrite the value.

Parameters: none

Expected Payload: The UISettings data object to store.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
201	OK, returns a JSON document.
401	not logged in. In this case anApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/globalSettings/{id}

Deletes the global settings with the given id.

Parameters:

Parameter	Description
id	The id of the settings to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return Object ofUISettings.
401	Not logged in. In this case anApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

8.2.52 /api/v1/userSettings

This API endpoint is used to save or get user settings for the users.

Supported HTTP methods: GET, PATCH, HEAD, OPTIONS

GET /api/v1/userSettings

Gets the saved settings in data base if not Backend should retrun default values with setting keys.

Parameters: none

Expected Payload: none.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
200	OK, returns a JSON document.
401	not logged in. In this case anApiError document is returned,describing details of the error.

PATCH /api/v1/userSettings

Save user setting in the Database.

Parameters: NONE

Expected Payload: The UISettings data object to store.

Replied Payload: The list of stored and default settings as an array ofUISettings data objects.

Return Codes:

Code	Description
201	OK, returns a JSON document with all user settings.
401	not logged in. In this case anApiError document is returned,describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/userSettings/{id}

Deletes the user settings with the given id.

Parameters:

Parameter	Description

Parameter	Description
id	The id of the settings to delete

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return Object of UISettings.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

DELETE /api/v1/userSettings

Deletes the user settings with username. Username is take from token by the backend.

Parameters: none

Expected Payload: none.

Replied Payload: none.

Return Codes:

Code	Description
202	ACCEPTED, Return list of UISettings array list.
401	Not logged in. In this case an ApiError document is returned, describing details of the error.
404	No user setting with this username exists . In this case anApiError document is returned, describing details of the error.
500	DB is not writable. In this case anApiError document is returned,describing details of the error.

8.3 WebSocket Communication

The API provides a websocket at /msgs to update volatile data asynchronously. The data to be sent over the websocket is controlled with a subscription model. With the sat-nms Software, everything is about device parameters: Every device parameter has a unique ID, changes of this parameters are transported as messages having the id and the value of this parameter as properties. The websocket is used one-way: There are only messages sent from the backend server to the front end. The /api/v1/subscribe API call controls which messages / parameters shall be send over the websocket.

8.3.1 Creating WebSockets

Creating a websocket id a two-step procedure: First a websocket ID must be obtained. This is done using the /api/v1/makewebsocket API call. This call prepares a websocket which can be connected to. It returns an ID for the newly prepared websocket. The client must remember this ID during the session, it is required for all subsequent operations on this websocket.

The second step is to connect to this websocket. The URL for these websockets is **/msgs**, all connections share this. To identify which websocket to open, a query parameter defining the websocket ID must added to the HTTP URI when opening the websocket:

```
/msgs&satnmsWebsocketId=XXXXX
```

XXXXX has to be replaced by the 5-character websocket ID obtained with the /api/v1/makewebsocket API call.

The now connected websocket does not receive any data.

8.3.2 Subscribing For Messages

To receive data over the websocket, the data must be subscribed for. With the sat-nms Software, each device parameter and each type/range description has its unique message ID. For each message ID an explicit subscription must be done to get this parameter and its updates. The /api/v1/subscribe API call does this, either for a single message ID or for a list of IDs. Subscriptions are made for a certain websocket instance, hence the ID of the websocket must be supplied with the subscribe call.

Subscribed messages are sent asynchronously, each time when a change of the parameter is detected. It is not guaranteed, that a subscribe results immediately in a message sent over the websocket. Parameters may be in the state "unknown", specially in the first minutes after starting a M&C. Such a parameter will be sent when the first value becomes available.

The /api/v1/subscribe API call is also is used to unsubscribe messages using the DELETE HTTP method. It is recommended to unsubscribe messages for which there are no more updates needed.

Closing the websocket automatically unsubscribes all pending subscriptions. There is no need to do this explicitly before closing the session.

8.3.3 Interpreting WebSocket Data

The websocket transports text messages containing single JSON data objects. These objects all are Message objects. Parsing and interpreting the Message objects can be done along the*type* field in the message. The value of this field determines the data type and further content of the message:

Token	Message Type
VALUE	Parameter, contains an update for a parameter value
RANGE	Range, contains a data type an valid range definition for a parameter
PONG	This could be an echoed message used by the front end as a heartbeat (see Echo / Heartbeat Function below)

Range messages contain an additional *dataType* field which specifies the data type of the parameter belonging to this range definition. Depending on *dataType*, there will me more fields in the range message, specifying the valid range of a numeric parameter, the available choices for an enumeration and more. The table below list all defined data types:

Token	Data Type
INTEGER	IntegerRange
HEX	HexRange
FLOAT	DoubleRange
BOOLEAN	BooleanRange
ALARM	AlarmFlagRange
CHOICE	EnumRange
OBJECT	ObjectRange

8.3.4 Echo / Heartbeat Function

"Ping" messages on websocket protocol level sent to the websocket are answered with a "Pong" message. This can be used to implement a heartbeat mechanism at protocol level.

At application level this is possible as well. The backend's websocket implementation echoes all text messages received from the front end back to it. It is recommended to use Message data objects for this and setting *type* to a value not used by the backend.

8.3.5 Closing Websockets

When a websocket is closed, the backend server deletes all subscriptions made for this websocket. The websocket cannot be re-opened, the websocket ID obtained for the closed

websocket becomes invalid.

When there is no websocket opened in a session for 120 seconds, the backend server deletes the session, this is the way the backend server handles the case when the operator closes the browser window. *(not yet implemented)*

8.4 Debug Terminal Communication

The backend provides a relay mechanism to access the debug terminal at each M&C connected to the backend. While sending commands to the debug terminal is done with an API call, the messages showing in the terminal are sent to the front end through a websocket. The following paragraphs give an overview about this.

8.4.1 Opening A Debug Terminal

To open a debug terminal, the front end opens a websocket at the URI defined for this. It receives every line to be shown in the terminal window as one message on the websocket. The M&C allows only one client to connect to the debug terminal at a time, opening the websocket may fail if either the M&C is down or if another user is already connected to the debug terminal (either from the Web-UI or from a Java client).

The URI to connect to the debug terminal is `/debug&mncName=XXXXX`

Where XXXXX has to be replaced with the name of the M&C to connect to. Once opened, the websocket transports every line from the debug terminal as a DebugMessage data object to the front end.

8.4.2 Closing A Debug Terminal

While the websocket for a debug terminal is opened, this prevents other users from using this feature. Hence every debug terminal websocket should be closed by the front end as soon as it is no longer needed. Closing the websocket also closes the data connection the M&C debug terminal port and by this frees this resource.

8.4.3 Sending Debug Commands

Sending commands to the debug terminal is done with the/api/v1/debug API call. This also takes a DebugMessage data object payload, carrying the line to be entered at the debug terminal. /api/v1/debug API calls require a debug terminal websocket to the referenced M&C being already opened. The will fail if the websocket for this M&C is not open because this means that also the data connection to the M&C is not present.

8.4.4 M&C Redundancy Management

The backend provides a redundancy switching mechanism for M&C systems connected to it. A redundant pair of M&C systems share the M&C name, but the M&C systems have different IP addresses. The redundancy works asymmetrically (there is one primary and one backup M&C) and one-way (redundancy switch will never happen from backup to primary).

If the communication to the primary M&C breaks and cannot be re-established for some time, the backend stops the M&C service on the primary machine and starts it on the backup machine. From now on the backend connects to the IP of backup M&C to control it.

All configuration about this is done in the 'backend.properties' file. The second IP address for the backup M&C and parameters how long the backend shall retry to connect to the primary M&C before it does the switch over to the backup M&C.

The actual state of the redundancy for each M&C (pair) is reported through the /api/v1/mnclist API call. Each MncInfo record in the returned list defines a 'redundancy' field which tells about the actual state.

8.4.4.1 Redundancy Commands

While the general aspects of the redundancy are statically configured in the 'backend.properties' file, the backend provides a /api/v1/redundancy API call to control if the primary or the backup machine of a redundant pair shall be used. There are two commands which may be sent with this API call.

command	description
RESET	tells the backend to switch back to primary M&C. If the backend is already connected to the primary M&C, or if no redundancy is configured for this M&C, the 'RESET' command has no effect. If the backend cannot connect to the primary M&C for some time, it tries to switch over to the backup machine.
FORCE	forces a switch over to the backup M&C - even if the communication with the primary one works fine. Has no effect if no redundancy is configured or if the backend is already connected to the backup machine.

8.5 Data Models

This section describes the data models used as payload with the API calls mentioned above.

The data model descriptions appear in an order which groups related functions. An alphabetic list of all data models is listed below

- ActiveBackend
- AcuTarget
- AcuTargetList
- AlarmFlagRange
- ApiError
- ArrowElement
- AzElElement
- BackendInfo
- StreamKeyData
- BooleanRange

- ButtonElement
- ChannelData
- ChartElement
- ColorDefinition
- DatabaseVersion
- DebugMessage
- DeviceDefinition
- DeviceElement
- DeviceFrameDefinition
- DeviceFrameTab
- DevicePreset
- DeviceSetup
- DeviceThreadDefinition
- Dictionary
- DisplayElement
- DocumentItem
- DocumentList
- DoubleRange
- EditButtonElement
- EnumRange
- EventAck
- EventInfo
- EventLog
- EventReportRequest
- EventReport
- FaultElement
- FaultInfo
- FaultList
- FrameElement
- FRViewProperties
- FRViewTraceDescription
- FRViewPreset
- FRViewTraceScaling
- GaugeElement
- HexRange
- IconElement
- IntegerRange
- InventoryDevice
- InventoryItem
- InventoryLogEntry
- ItemList
- KeyValPair
- LatchingButtonElement
- LockButtonElement
- MCPElement
- Macro
- Message

- MncInfo
- MncList
- ObjectRange
- ParameterButtonElement
- ParameterElement
- Parameter
- PresetValue
- PresetVars
- DeviceVars
- RadioButtonElement
- Range
- RectElement
- RedundancyCmd
- RedundancyState
- RedundancySummary
- RestartCmd
- SatDbAntenna
- SatDbBeacon
- SatDbBeaconAttenuation
- SatDbI11Data
- SatDbPosition
- SatDbSatDetails
- SatDbSatOperator
- SatDbSatellite
- SatDbTLEData
- SatDbTableTracking
- SatDbTc
- SatDbTcAttenuation
- SatDbTleImport
- SatDbTm
- SatDbTmAttenuation
- SatDbState
- ScheduleEvent
- Schedule
- ScreenDefinition
- ScreenElement
- SpectrumElement
- SpectrumTrace
- StringRange
- SubscribeList
- SwitchElement
- TakeButtonElement
- TargetListElement
- TextElement
- Thumbnail
- TokenReply
- TrackingHistory

- TrackingPoint
- TreeViewNode
- UISettings
- UserList
- UserProperties
- WebsocketId
- XYChartElement

8.5.1 TokenReply

The TokenReply data model contains the data returned from a successful login or token refresh (API end point /token). The fields defined in this data model follow the OAuth2 recommendations.

Data Model

Key	Type	Value
access_token	String	The access token (JWT) to be added as 'Authorization: Bearer ...' header field with subsequent API calls.
refresh_token	String	The refresh token (JWT) to be used to refresh an expired access token with the /token API end point and *grant_type* set to 'refresh_token'.
token_type	String	Always 'JWT'
expires	Number	The number of seconds the access_token is valid

8.5.2 UserProperties

The UserProperties document contains some information about the user currently logged in. It is returned as a reply to the GET /api/v1/user call.

Data Model

Key	Type	Value
userName	String	The user (login) name.
fullName	String	The user's full name.
password	String	The user's password (encrypted).
groupList	String	A comma separated list of group identifiers this user is allowed to access.
privilege	Number	The user's privilege level (1..150)

Remarks

- The user's privilege level is very important for the front end as it specifies which user interface elements may be accessed by the user. With the sat-nms WebGUI the privilege level mechanism replaces the commonly used 'roles'. This works the following way:
  - Each active user interface element like an entry field, a drop down box has a privilege level defined. Predefined screens like standard device screens have these level set to standard values, with user defined screens each user interface element may be configured to an individual privilege level.
  - If the privilege level of the user is higher or equal to the level of the user interface element, he may change the parameter controlled by this element.
  - If the privilege level of the user is lower, he gets the actual value of the parameter displayed, but he may not change this value.
- The privilege level has a range from 1 to 150, commonly used values are 100 for standard operator access and 150 for admin privileges.

8.5.3 UserList

The UserList document contains the complete list of users known to the backend. It is returned as a reply to the GET /api/v1/users call.

Data Model

Key	Type	Value
users	Array	Contains the list of known users as an array ofUserProperties objects

8.5.4 ApiError

The backend server returns an ApiError document if something went wrong, i.e. The HTTP return code is not in the 200 range.

Data Model

Key	Type	Value
timestamp	String	The time and date when the event happened.
status	Number	The HTTP status code returned.
error	String	The textual description of *status*.
message	String	A message describing the reason of the error.
path	String	The path part of the URL which caused the error.

Remarks

- The *timestamp* has a format like `2020-01-16T09:40:48.079+0000`

- The fields *error* and *message* contain the same text in most cases.

8.5.5 MncList

The MncList is used with the /api/v1/mnclist API call to report the list of M&C systems managed by the backend.

Data Model

Key	Type	Value
mncList	Array	contains an array of MncInfo objects, each describing the properties of one M&C system manged by the backend. The first entry of the list is called the primary M&C, it is used to get the login data and the live event log. This (and only this) M&C has "primary" set true.

8.5.6 MncInfo

The MncInfo data model contains the properties of one entry in a MncList. It is used with the /api/v1/mnclist API call to report the list of M&C systems managed by the backend.

Data Model

Key	Type	Value
id	String	The ID / name of the M&C. This ID is used to identify the M&C in messages. See section Message below for details how the M&C ID is embedded in the message ID.
ipAddress	String	The IP address of the M&C server as defined in the backend configuration file.
connected	Boolean	The actual state of the connection to the M&C.
primary	Boolean	True marks this M&C as the primary one.
redundancy	String	Describes the state of the redundancy control for this M&C. Possible values are 'NONE' = no redundancy control configured for this M&C 'PRIMARY' = the backend connects to the primary IP address for the M&C 'BACKUP' = the backend connects to the backup IP address for the M&C

8.5.7 ActiveBackend

The ActiveBackend data model carries the name of the backend which actively controls M&C redundancy switching in a redundant backend configuration. It is used with the /api/v1/activebackend API call.

Data Model

Key	Type	Value
activeBackendName	String	The name of the backend which is or shall be the active backend

The name of a backend is defined in the 'backend.properties' file with the 'backend.name' key.

8.5.8 SubscribeList

The SubscribeList is used with the /api/v1/subscribe API call to define a list of message identifiers the front end wants to subscribe or to unsubscribe from.

Data Model

Key	Type	Value
subscribeList	Array	contains an array of strings, each string is on message ID to be subscribed / unsubscribed.

8.5.9 WebsocketId

The WebsocketId carries the ID of a newly created websocket instance. It is returned by the /api/v1/makewebsocket API call in the response body.

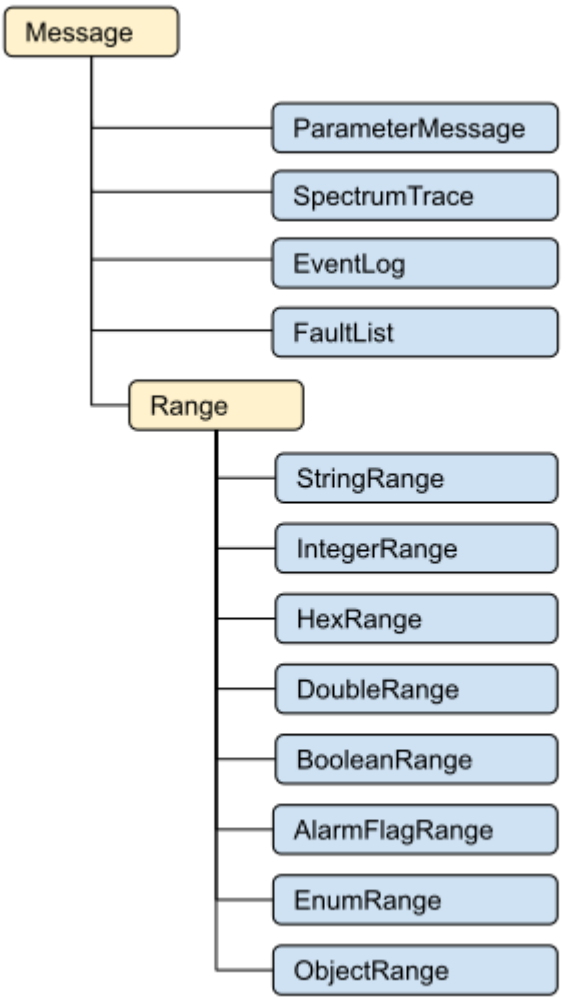
Data Model

Key	Type	Value
id	String	A newly created websocket ID. This is a unique 5-character string identifying the websocket for all further processing.

8.5.10 Message

With the sat-nms software, all volatile data gets exchanged as a Message. A message may carry a parameter like a value read from a device or a setting for this device. Beside this, data type and range information about parameters are exchanged as messages as well.

Because of this there are several flavors of messages defined, based of a hierarchical model derived from object oriented programming. The diagram below shows the known variants of Message as they depend on each other. Data models with yellow background do not exist in reality, they are defined as 'abstract base classes' which define the common parameters of their siblings.



Messages are used either with the /api/v1/peek and /api/v1/poke API calls or with the subscription model over a websocket connection.

The core property of a Message is its message ID. The message ID describes, which parameter of which device connected to which M&C server is addressed with the information of the message. The general format of a message ID is:

```
{M&C-Name}.{Device-Name}.{Parameter-Name}[.R]
```

The message ID is composed of at least three words with dots in between. The first word describes the M&C server to which the device is connected which is addressed by the second word. The third word identifies one parameter of this device, this identifier again may contain dots, if the parameter in the device are organized in a tree-like structure. Finally, the character sequence **.R** may be appended to the ID. This marks the message containing the range information about the parameter instead of the actual parameter value. Example:

```
MNC-1.DEVICE-3.parameter
```

Data Model

Key	Type	Value
id	String	The message ID associated with this message.
type	String	The message type (see below).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ' (UTC)

Defined Message Types

Token	Data Type
VALUE	A Parameter message.
RANGE	A Range message.
SPECTRUM	A SpectrumTrace message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
EVENTLOG	An EventLog message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
FAULTLIST	A FaultList message. This message may not be sent with a /api/v1/poke call, it is read-only from the front end's point of view.
PONG	Echo reply (PONG is only an example, the front end may send any message type other than *VALUE* or *RANGE*, the backend will echo the message as it is received.)

Message IDs of Range messages always end with **.R** - this is the way the underlying ***sat-nms*** M&C software marks these messages.

8.5.11 Parameter

A Parameter message carries a value to or from a device controlled by the M&C server. When sent from the UI to the M&C server, the Parameter message commands a new value to be set at the device. When sent from the M&C server to the UI, the message informs about a change of the parameter. This applies to read-only parameters like state flags or measurement values and to settings-type parameters as well.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message type (always **VALUE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
val	String	The value of the parameter, coded as a String.

The value contained in a Parameter message (*val*) may be of different types. The Range of the parameter defines this type, strictly spoken the *dataType* field in the Range message. For simplicity, all data types are coded as a string value in the Parameter message. The table below specifies, in which way parameter values are coded for all defined *dataType* flavors:

Range *dataType*	Coding in the *val* field
TEXT	UTF8 coded String.
INTEGER	Decimal number, consisting of digits 0..9
HEX	Decimal (!) number, consisting of digits 0..9
FLOAT	Floating point number with decimal point
BOOLEAN	Either **true** or **false** (all lower case)
ALARM	Either **true** or **false** (all lower case)
CHOICE	The selected choice as a UTF8 coded string. Please note, case is significant with choices. The values **ON** and **On** are treated as different ones.
OBJECT	ObjectRange parameter are not supported with this version of the API. Writing an ObjectRange parameter to the M&C is ignored, when queries an ObjectRange parameter always reports an empty string in *val*.

8.5.12 Range

A Range message contains the type definition and an optional specification for a range/limit validation for numeric parameters. The data model definition for Range is abstract, which means it never will be used in practice, only the data models derived from this will be used in the API.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	A token defining the data type this Range object describes. Depending on *dataType*, there are more parameters present in the Range definition.
disabled	Bool	True tells the front end that this parameter is currently not in use. No value shall be displayed for this parameter, even if a corresponding Parameter message has been received. Any decorations like frame or descriptive label shall be displayed grayed/faded in order to show this state to the operator.
readonly	Bool	This parameter is for display only, the operator may not change it.

Defined Data Types

Token	Data Type
TEXT	StringRange
INTEGER	IntegerRange
HEX	HexRange
FLOAT	DoubleRange
BOOLEAN	BooleanRange
ALARM	AlarmFlagRange
CHOICE	EnumRange
OBJECT	ObjectRange

Remark: Range objects are also used in the PresetVars list of device driver variables. In this case *messageId* does not contain a fully qualified message ID, but the name of the variable.

8.5.13 StringRange

A StringRange parameter carries a free text.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)

Key	Type	Value
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **TEXT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

8.5.14 IntegerRange

An IntegerRange parameter carries an integer number. Internally the value is represented by a signed 64-bit number (data type *long*).

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **INTEGER** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

8.5.15 HexRange

A HexRange parameter carries an integer number. Internally the value is represented by a signed 64-bit number (data type *long*). HexRange is much like IntegerRange with the only difference, that values are printed at the UI in hexadecimal notation (no prefix like '0x' or '#') and user entries are expected in the same format.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **HEX** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

8.5.16 DoubleRange

A DoubleRange parameter carries a floating point value. Internally floating point values are represented by 64-bit floating point values (data type *double*, that gave the name for the DoubleRange).

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **FLOAT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

Key	Type	Value
checkLimits	Boolean	true = the front end must validate any entry for this parameter to be in the range *min ... max*. Invalid entries must be rejected, no values shall be sent to the server if outside the given range.
min	Number	The minimum value to be accepted with *checkLimits*=true
max	Number	The maximum value to be accepted with *checkLimits*=true
prec	Number	The precision (number of digits following the decimal point). Details are given in the paragraph **Number Formatting** below
unit	String	A unit string to be displayed right of the parameter value (e.g. "MHz" for a frequency)

Number Formatting

FLOAT variables are displayed with a fixed precision. The *prec* parameter defines how many digits right of the decimal point shall be displayed. Values of *prec* above 100 define a scientific formatted floating point with *prec - 100* digits precision (e.g. '0.123E-2' for *prec=103*)

8.5.17 BooleanRange

A BooleanRange parameter carries a BOOL value. It may be one of the values *true* or *false*.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **BOOL** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

8.5.18 AlarmFlagRange

The AlarmFlagRange describes a sat-nms ALARM parameter. Like the BOOL parameter may be *true* (alarm) or *false* (no alarm). The AlarmFlagRange defines a descriptive text for the alarm flag which can be displayed beside the actual state.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **ALARM** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range (always *true* with alarm flags)
description	String	A short description of the ALARM

8.5.19 EnumRange

The EnumRange describes a parameter which' value can be one of a given set of values. An UI typically implements this parameter type as a drop down box or as a list box which allows a single selection.

Data Model (extends Range)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **CHOICE** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range
choices	Array	An array of string containing the choices for this parameter

8.5.20 ObjectRange

An ObjectRange describes a parameter carrying binary data, e.g. A spectrum analyzer trace. Special software is required to decode the data contained.

Data Model (extends Range)

Key	Type	Value

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message data type (always **RANGE**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
dataType	String	Always **OBJECT** (see Range)
disabled	Bool	Inherited from Range
readonly	Bool	Inherited from Range

8.5.21 SpectrumTrace

A SpectrumTrace message contains the spectrum trace data distributed by a CSM-Spectrum-Analyzer device. The spectrum trace is much like a read-only parameter (it may not be commanded to the M&C), instead of a single value it carries an array of numeric values.

The array elements represent the spectrum trace, all elements are integer numbers describing the point position in pixels from the top of the display downwards. The length of the trace depends on the particular spectrum analyzer model, common trace lengths are 501 or 601 points. The first array element is the leftmost point (at *centerFrequency-frequencySpan/2*), the last array element is the is the rightmost point (at *centerFrequency+frequencySpan/2*).

The point positions in the array (array index, element value) refer to the native size of the spectrum analyzer display as read out by the driver. This spectrum display size in pixels may be read from the spectrum analyzer device as the 'info.screen' variable. The spectrum analyzer widget on the screen may however re-scale the display to any size requested.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID (see Message)
type	String	The message type (always **SPECTRUM**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
trace	Array of Number	The spectrum trace, as an array of integer values, describing the point position in pixels from the top of the display downwards.

8.5.22 TrackingPoint

A TrackingPoint message contains pointing data of an antenna tracking controller for one tracking step. An array of TrackingPoint elements makes up the TrackingHistory which is

described below

Data Model

Key	Type	Value
time	String	A time stamp telling when this tracking point was recorded. The format follows ISO 8601, example: `2021-03-26T08:02:30Z`, time zone is always UTC.
apos	Number	The azimuth pointing of the antenna after the tracking step.
epos	Number	The elevation pointing of the antenna after the tracking step.
blev	Number	The beacon level after the tracking step.
asucc	Boolean	True = the peaking process of the azimuth axis was successful.
apeak	Number	The azimuth peak position evaluated during the tracking step. Only meaningful if *asucc* is true.
esucc	Boolean	True = the peaking process of the elevation axis was successful.
epeak	Number	The elevation peak position evaluated during the tracking step. Only meaningful if *esucc* is true.

8.5.23 TrackingHistory

A TrackingHistory message contains the tracking memory of an antenna tracking controller. Actually, only the SatService ACU provides this data. The TrackingHistory message is read-only, meaning it may not be commanded to the M&C.

Data Model

Key	Type	Value
trace	Array of TrackingPoint	The tracking memory. The tracking points are ordered by time in the array, the first element refers to the oldest tracking point.

8.5.24 EventLog

An EventLog message carries the up to 24 most recent events that occurred in one of the M&C systems managed by the backend. The backend collects the events from the M&C systems and combines them to this list. The backend updates this message whenever a new message is added to the log, but at a maximum rate of one message per second.

To subscribe for the EventLog message, a/api/v1/subscribe API call must be made for the message ID `{Primary-M&C-Name}.SYSTEM.eventLog` where {Primary-M&C-Name} is the name / ID of the M&C reported with the /api/v1/mnclist API call with the "primary" key set true.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID. This is always the message ID described above. (see Message)
type	String	The message type (always **EVENTLOG**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
events	Array	An array of EventInfo objects representing the recent events which occurred. This array has not more than 24 elements, directly after system start the array may be shorter or even empty if no events have been collected yet.

8.5.25 EventReport

An EventReport contains a list of EventInfo objects. The EventReport is the reply of the backend for a /api/v1/eventreport call when sent as a POST.

Data Model

Key	Type	Value
events	Array	An array of EventInfo objects.

8.5.26 EventReportRequest

The EventReportRequest specifies the filters to be applied when generating an EventReport. It has to be sent with an /api/v1/eventreport POST call.

Data Model

Key	Type	Value
from	String	Specifies the start of the time range of the report. Must be formatted following ISO 8601, example: `2021-03-26T08:02:30Z`, time zone is always UTC.
to	String	Specifies the end of the time range of the report. Must be formatted following ISO 8601, example: `2021-03-26T08:02:30Z`, time zone is always UTC. May be *null* if *length* is not zero.
length	Number	The length of the time range of the report in hours. If this value is zero, the report length is determines as the difference *to* - *from*. If the value is not *null*, the *to* value is ignored and *length* is used as the report length.

Key	Type	Value
mnc	String	The M&C name filter. Only events containing this text in their *source1* (M&C name) field are included in the report. If*mnc* is *null*, all events pass this filter.
device	String	The device name filter. Only events containing this text in their *source2* (device name) field are included in the report. If*device* is *null*, all events pass this filter.
host	String	The host name filter. Only events containing this text in their *origin1* (host name) field are included in the report. If *host* is *null*, all events pass this filter. Please note that only parameter change events have a host name set, other events will never match this filter.
user	String	The user name filter. Only events containing this text in their *origin2* (user name) field are included in the report. If *user* is *null*, all events pass this filter. Please note that only parameter change events have a user name set, other events will never match this filter.
maxlen	Number	The maximum length of the report (number of event messages). If more events than this number match the filter settings, only the oldest *maxlen* events of these will be contained in the report. If maxlen is null, the server side default for the maximum report length is used.
pri	String	The event priority filter. Must contain one of the following: ALL = include all events (no filter) INFO = include the events with priority INFO or higher INFO-ONLY = include the events with exactly priority INFO WARNING = include the events with priority WARNING or higher WARNING-ONLY = include the events with exactly priority WARNING FAULT = include the events with priority FAULT or higher FAULT-ONLY = include the events with exactly priority FAULT
text	String	The message text filter. Only messages which contain the given text are included in the report. If *text* consists of multiple words separated by space characters, the message text must contain all these words in arbitrary sequence. If *text* is *null*, all events pass this filter.

8.5.27 EventAck

The EventAck tells the backend which events in the event log shall be acknowledged. It is sent with a POST call to /api/v1/eventack.

Data Model

Key	Type	Value
ackAll	Boolean	Selects if a single event shall be acknowledged (false) of if all pending events shall be acknowledged (true).
event	EventInfo	The event to be acknowledged. *event* is ignored if *ackAll* is true.

8.5.28 EventInfo

An EventInfo contains the properties of one event within an EventReport or an EventLog message.

Data Model

Key	Type	Value
id	Number	A unique numeric identifier for this record. The event id is required to acknowledge a particular event.
pri	Number	The event priority coded as a one digit number 0 = suppressed, the event shall not be shown 1 = informational 2 = fault 3 = alarm 4 = warning
nak	Boolean	*(needs to be acknowledged)* This reads true for unacknowledged fault events.
msg	String	The event message text.
t1	String	*(time1)* A time stamp defining the time when the event was detected. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
t2	String	*(time2)* A time stamp defining the time when the event was stored in the event database. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
s1	String	*(source1)* The name / id of the M&C which generated this event.
s2	String	*(source2)* The name of the device which generated this event.
o1	String	*(origin1)* The name of the originator. Used with parameter change events to tell who commanded the change. May be a user name or in case of automatic parameter changes the name of the device which caused the change.
o2	String	*(origin2)* The host name / IP address from where a parameter change was commanded.

8.5.29 FaultList

A FaultList message carries the list of active faults and warnings which are pending at a given M&C. The backend updates this list whenever a fault changes at this M&C but at a maximum rate of one message per second.

To subscribe for the FaultList message, a/api/v1/subscribe API call must be made for the message ID `{M&C-Name}.SYSTEM.faultList` where {M&C-Name} is the name / ID of the M&C of interest.

Data Model (extends Message)

Key	Type	Value
id	String	The message ID. This is always the message ID described above. (see Message)
type	String	The message type (always **FAULTLIST**, see Message).
time	String	Timestamp in format 'YYYY-MM-DDTHH:MM.SSZ', see Message).
faults	Array	An array of FaultInfo objects representing list of actually pending faults and warnings at the given M&C.

8.5.30 FaultInfo

A FaultInfo contains the properties of one fault of warning reported in aFaultList message.

Data Model

Key	Type	Value
time	String	A time stamp telling when this fault changed its state. The format follows ISO 8601, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
src	String	The name of the device which generated this event.
pri	Number	The event priority coded as a one digit number 0 = suppressed or device out-of-service, the fault shall not be shown 1 = informational 2 = fault 3 = alarm 4 = warning
msg	String	The fault message text.

8.5.31 DeviceFrameDefinition

A DeviceFrameDefinition document defines tabs to be shown in a sat-nms device screen. It resembles the content of a file in the *dframes* directory of the ***sat-nms*** M&C software.

DeviceFrameDefinition documents are returned by a GET /api/v1/dframes/{name} call.

Data Model

Key	Type	Value
name	String	The file name of the dframe definition
allowPresets	Boolean	true = the UI shall show the load/save preset buttons for this dframe
presetPrivilege	Number	the privilege level required to launch the preset load/save dialogs. This only is meaningful with *allowPresets=true*
tabs	Array	Contains a list of DeviceFrameTab objects, each describing one tab of the device frame

8.5.32 DeviceFrameTab

A DeviceFrameTab document defines one tab to be shown in a***sat-nms*** device screen. It appears as part of the DeviceFrameDefinition data object.

Data Model

Key	Type	Value
icon	String	The name of the icon to be shown in the tab header
dscreen	String	The name of the dscreen to be shown in this tab
description	String	A description of what is shown in this tab, the Java UI shows this as a mouse over tooltip

8.5.33 ScreenDefinition

A ScreenDefinition document defines the layout of a screen / page to display. The document contains some global properties of the screen and an array of ScreenElement objects, which each define one UI widget with its properties.

With the sat-nms software there are two general types of screen definitions: 'user defined screens' and 'device screens'. While the first type may contain widgets which show parameters of any device controlled by the sat-nms M&C are 'device screens' bound to one particular device. Device screens are returned by a GET /api/v1/dscreens/{name} call, user screens by a GET /api/v1/uscreens/{name} call.

Data Model

Key	Type	Value
name	String	The file name of the screen definition
deviceRelative	Boolean	true = contains no device IDs, the front end application must add the device ID from the context of the screen
width	Number	Width of the screen (pixels in the Java client)
height	Number	Height of the screen (pixels in the Java client)
bgImage	String	Name of a background image to draw in the screen. An empty value states that there is no background image.
noEditTake	Boolean	true = exclude this screen from global edit/take if this feature has been set for the UI.
elements	Array	Contains ScreenElement objects, the list of the screen elements contained in this screen.

8.5.34 ScreenElement

A ScreenElement object describes the properties of one UI widget contained in a ScreenDefinition. This comprises the type of widget, its location and size and many more parameters which depend on the type of widget this ScreenElement defines. This way ScreenElement is kind of an abstract definition or 'base class' in terms of object oriented programming. ScreenElement defines the parameters which are common to all types of screen elements.

Data Model

Key	Type	Value
type	String	The type of the screen element. See the list of screen element types below for details.
xpos	Number	X-position of the upper left corner of the element in the screen (pixels in the Java client)
ypos	Number	Y-position of the upper left corner of the element in the screen (pixels in the Java client)
width	Number	Width of the element (pixels in the Java client)
height	Number	Height of the element (pixels in the Java client)

As mentioned above, a ScreenDefinition will never contain a ScreenElement with only the parameters shown in the table above. All ScreenElement objects will be of a certain screen element type and will contain all parameters defines for this element type. The parameters described for ScreenElement will be always included.

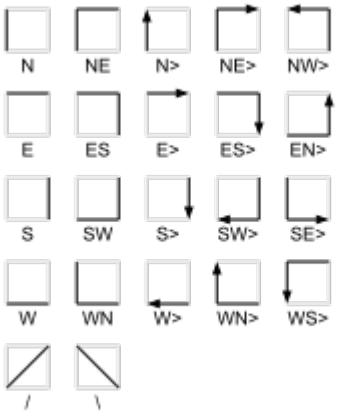
List of screen element types:

Element Type	Description
ArrowElement	Draws a horizontal/vertical line or arrow.
AzElElement	A chart showing the tracking history of an antenna controller as a point cloud in an elevation over azimuth coordinate system.
ButtonElement	Defines a button which launches another screen, e.g. for detail views.
ChartElement	A strip chart element, displaying a numeric parameter as an y/t diagram that automatically advances with 1 pixel / second.
DeviceElement	Places an icon into the screen which represents a device. This icon will display the device's operating/fault state by it's color / shape.
DisplayElement	A parameter display field which does not allow the parameter to be edited.
EditButtonElement	Places an EDIT button into the screen, implicitly forcing the EDIT/TAKE operating method for this screen instead of changing each parameter independently.
FaultElement	A special display element to display a fault flag. It shows the textual description of the fault and it's state in an entry field frame. The right mouse button shows a context menu to change the fault's priority
FrameElement	Draws a sunken 3D frame, may be used to group parameters.
GaugeElement	A gauge element, displaying a numeric parameter as a horizontal bar.
IconElement	Places an icon (PNG, GIF or JPG image) into the screen. Optionally the icon can be programmed to change with a parameter value.
LatchingButtonElement	A button which displays/controls a 2-state parameter using it's pressed state.
LockButtonElement	A button to control the lock state for exclusive operation for a number of devices.
MCPElement	The MCP display element integrates the measurement display of a spectrum analyzer device in multi channel measurement mode in a user screen.

Element Type	Description
ParameterButtonElement	A button which sends a certain parameter value when pressed.
ParameterElement	Either a parameter entry field or a drop down box, depending on the type of parameter.
RadioButtonElement	A parameter entry field specially for CHOICE parameters.
RectElement	Draws a rectangle.
SpectrumElement	The spectrum display element integrates the spectrum display of a spectrum analyzer device in a user screen.
SwitchElement	Like the device icon, but additionally displays the actual position of a switch (Meant to be used for block diagrams showing the true signal path.
TakeButtonElement	Places a TAKE button into the screen.
TargetListElement	The ODM Target List element shows the list of targets of a SatService-ACU-ODM antenna controller device. It permits to recall, save or delete target definitions of this type of antenna controller. It is specialized to this antenna controller, does not support other types.
TextElement	Displays a single line of text.
ThumbnailElement	Displays a thumbnail image fetched directly from a video processing device (encoder, decoder, gateway)
XYChartElement	This element shows the relation of two numeric variables in an X/Y diagram, featuring a trace which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

8.5.35 ArrowElement

The ArrowElement draws a horizontal and/or a vertical line and optionally an arrowhead. The line's color is selectable, also may follow a parameter value. Strictly spoken the ArrowElement is a rectangle with only one or two sides drawn. The parameter *arrowCode* defines which sides of the rectangle shall be drawn and if an arrow shall be placed at the end of the line. The figure below shows all arrow codes defined:



Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ArrowElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. Is only relevant if *variableLineColor* reads *true*.
arrowCode	String	A 1..3 character code defining the line/arrow outline. See the figure above which arrow codes are possible.
color	String	The color used for the line / arrow.
variableLineColor	Boolean	true = set the line color from the value received from the parameter addressed by *id*.
useColorTranslation	Boolean	use the colors[] translation
colors	Array	of ColorDefinition objects. Defines how to translate values of the *id* parameter to color values if *useColorTranslation* reads *true*.

Variable Line Color

With *variableLineColor=true* the ArrowElement will change its line color depending on the value of the parameter addressed by *id*. This way the line can be used as a status indicator, e.g. in a block diagram signalling "the signal goes actually this way". The way the ArrowElement interprets the parameter value depends on the value of *useColorTranslation*.

If *useColorTranslation=false*, the ArrowElement tries to decode the line color directly from the parameter value. First it does the following tests on the parameter value (not case sensitive):

Parameter Value	**Line Color And Thickness**
ends with **BOLD**	The line is drawn thicker than normal
starts with **BLACK**	The line is drawn with this color
starts with **WHITE**	The line is drawn with this color
starts with **RED**	The line is drawn with this color
starts with **BLUE**	The line is drawn with this color
starts with **GREEN**	The line is drawn with this color
starts with **YELLOW**	The line is drawn with this color
starts with **GRAY**	The line is drawn with this color
contains **TRUE**	If the message ID identifies the parameter as an alarm flag (the ID contains 'fault'), the line is drawn **RED-BOLD**, for other parameters it is drawn **GREEN-BOLD**
contains **FAULT**	The line is drawn **RED-BOLD**
starts with **ON**	The line is drawn **GREEN-BOLD**

If none of the above conditions apply, the ArrowElement tries to interpret the parameter value as a hex coded color (#RRGGBB). If this also fails, the standard line color defined in *color* is used.

If *useColorTranslation=true*, the ArrowElement uses the ColorDefinition table in *colors[]* to decode the line's color and thickness.

8.5.36 ButtonElement

The ButtonElement creates a button which launches another user interface screen if pressed.

The screen to be launched may be another user defined screen or a predefined one.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
privilege	Number	The privilege level required to make a button click send a value
label	String	A label shown on the button
icon	String	An icon shown on the button
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
action	String	The action to perform if the button is clicked (see below)
par1	String	A parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
par2	String	A second parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
par3	String	A third parameter that applies for the action to perform. The meaning of this field depends on the *action* selected (see below)
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

The *action* field of the ButtonElement defines what the button will do when clicked by the operator. The list below shows all known values for *action*, the meaning of this and a description of the parameters *par1* and *par2* which will have different meanings for particular action.

CHILD-SCREEN

Launches a new browser tab or window showing the user screen referenced with *par1*.

- *par1:* The name of the user screen to show.
- *par2:* null
- *par3:* null

REPLACE-SCREEN

Replaces the actual user screen by the one referenced with *par1*.

- *par1:* The name of the user screen to show.
- *par2:* null
- *par3:* null

LOAD-PRESET

Launches a "Load Preset" dialog which lets the operator select and apply a device preset from a list of presets which are defined for this particular type of device. Device presets are local to each M&C instance, have to be loaded and applied from there.

- *par1:* The name of the device to which the preset shall be applied to. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYDEVICE` (must be set in the screen editor this way).
- *par2:* The name of the driver of this device.
- *par3:* An option search pattern to filter the list of displayed presets. All presets containing the given string (not case sensitive compare) shall be shown. If *par3* is null, all preset for the given device / driver combination shall be shown.

TRACK-VIEW

Launches a tracking view window. This window shows the tracking history of a given antenna controller device. Not all tracking controller devices are capable to report their tracking history.

- *par1:* The name of the antenna controller device for which the tracking history shall be shown. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYACU` (must be set in the screen editor this way).
- *par2:* null
- *par3:* null

AZEL-VIEW

Launches a Az/El view window. This window shows the tracking history of a given antenna controller device as a cloud of dots in a azimuth over elevation coordinate system. Not all

tracking controller devices are capable to report their tracking history.

- *par1:* The name of the antenna controller device for which the tracking history shall be shown. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MYACU` (must be set in the screen editor this way).
- *par2:* null
- *par3:* null

FREC-VIEW

Launches a File-Recorder view window. This window shows the data recorded by the File-Recorder device referenced with *par1* or the live data provided by this device.

- *par1:* The name of the File-Recorder device. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MY-FREC` (must be set in the screen editor this way). In the device screen of the file recorder, the special device name "." denotes *this* device.
- *par2:* The file recorder view preset number (0..7) to be applied. May be null, in this case no stored settings are applied.
- *par3:* null

SPECTR-VIEW

Launches a Spectrum Display window. This window in fact is a device screen of the CSM-Spectrum-Analyzer device.

- *par1:* The name of the spectrum analyzer device. The device name has the name of the M&C prepended where it is controlled. Example: `MYMNC.MY-FREC` (must be set in the screen editor this way).
- *par2:* A comma separated list of message-id/value pairs. Within each pair the message-id is separated from the value by one space character. Message ids are fully qualified, starting with the M&C name (must be set in the screen editor this way). The Spectrum Display window shall parse this list and send every message defined in the list with a poke call to the backend. This macro-like function is used to initialize the spectrum analyzer or to switch its input when the window is launched.
- *par3:* null

BROWSER-VIEW

Opens a new Browser window and displays a given URL in this window. This function is used to invoke the sat-nms online help and to launch the Web-GUI of certain devices.

- *par1:* The URL to show. Before launching the browser window, some replacements have to be done on the URL string:
  - If *par1* is a string consisting only of 4 decimal digits, this is a sat-nms online help topic number and the string must be expanded to the full URL from where this topic can be loaded by the browser.
  - If the ButtonElement is part of a device screen and the URL contains '@' characters, they have to be replaced with the (IP-) address of the device. A double '@@'

escapes this behavior and gets replaced by a single '@' in the URL string.

- *par2:* null
- *par3:* null

VIDEO-VIEW

Launches a VideoLAN video player to show the video stream from a surveillance camera controlled by a M&C. The VIDEO-VIEW ButtonElement may only occur in device screens, the IP address of the camera has to be taken from the address_ parameter of the device.

- *par1:* null
- *par2:* null
- *par3:* null

ROBOT-IMPORT

Launches a Robot Import screen. This screen lets the operator select a Pointing-Robot file from the local PC, the file gets converted to sat-nms format an then transferred to the M&C.

- *par1:* null
- *par2:* null
- *par3:* null

TREE-NAVIGATE

In a tree view UI this button navigates in the tree to the tree path referenced with*par1*. As the tree view UI with the WebUI is still t.b.d., this description should be considered as preliminary.

- *par1:* The tree view path to navigate to.
- *par2:* null
- *par3:* null

UNKNOWN

All ButtonElement definitions in a screen, which have no WebGUI equivalent for their programmed *action* (like "launching a new Java VM') are translated into an UNKNOWN action. The Frontend may show this e.g. as a disabled button with an "n/a" label.

- *par1:* The complete content of the message-id field in the original button definition.
- *par2:* null
- *par3:* null

8.5.37 ChartElement

The ChartElement shows a strip chart of a numeric parameter. The chart element keeps a local history of the received values, advances with a constant speed of 1 pixel / second. The default y-scale is 1/division but may be changed by clicking to the chart with the right mouse button.

By default, the strip chart element lets the y-scale offset follow the displayed value that the recent measurement samples are shown in the diagram. This behavior is optimized for

applications where the strip chart shall indicate a 'trend' for the displayed value, using an element height of only 50 pixels or less.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ChartElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. This should point to a numeric parameter to make sense.
label	String	The label shown above the strip chart
font	String	The font to be used for the label
color	String	The color used for the label and the strip chart line.
mode	String	The strip chart's display mode. This mainly controls the scaling behavior of the chart. See Display Modes below.
scale	double	The scale 1/div
minValue	double	The value corresponding the bottom line of the widget
maxValue	double	The value corresponding the top line of the widget
minThreshold	double	Values below this threshold cause the widget to display a fault
maxThreshold	double	Values above this threshold cause the widget to display a fault

Display Modes

The ChartElement uses different display and scaling modes, depending on the display mode defined in *mode*. The table below lists the defined display modes an their behavior.

mode	**Description**

mode	Description
FLOATING-AUTO	Does a full autoscale. The y-offset of the strip chart is set that the newest value displayed appers at the middle of the y axis. The chart y-scale is evaluated in a 1-2-5 raster to the finest scale that allows all points in the history to be displayed in the chart area. All scaling parameters are ignored.
FLOATING-FIXED	Evaluates the y-offset like in **FLOATING-AUTO**, but applies a fixed y-scale as defined in *scale*. The *scale* value is 1/div and chart height is assumed as 2 divisions. So, the top line of the chart corresponds to y-offset + *scale*, the bottom line to y-offset -*scale*.
FIXED	Sets a fixed y-range from the *minValue* / *maxValue* parameters.
FIXED-THRESHOLD	Like **FIXED**, but also checks every new value against the *minThreshold* / *maxThreshold* limits. If outside the limits, the chart background turns to red.
ANTENNA-AZEL	Makes the element to be a AZ/EL x/y diagram instead of the strip chart. The details about this element type will be defined in a later version of the API.

8.5.38 AzElElement

The AzElElement shows the tracking histoy of an antenna controller as a cloud of points in a elevation over azimuth coorsinate system. Actually only SatService antenna conrollers provide the tracking history data for this screen element.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "AzElElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name (prepended by the M&C name) of the antenna controller for which the AzElElement shall be shown.
label	String	The label shown above the chart
font	String	The font to be used for the label

8.5.39 DeviceElement

The DeviceElement represents a device in the M&C user interface. It displays the status of the device by it's color/shape and gives access to the device window for this particular device by a double mouse click. The right mouse button launches a context menu with common operations for the device.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "DeviceElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device ID this element represents in the UI. The device ID consists of the M&C name followed by a dot and the device name.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.
icon	String	The (base-) name of the icon to show for the device. See Icon Selection below.
showName	Boolean	true = show the device name als a label on top of the icon
showIsBackup	Boolean	true = use a different icon variant for this device if it is in 'is backup' state.
signalBackupId	String	A message ID to listen for a parameter from which the 'is backup' state is derived. Ignored unless *showIsBackup* is true.
backupValue	String	The parameter value which must be matched for the 'is backup' state.
font	String	The font to be used for the label.
color	String	The color to be used for the label (#RRGGBB)

Based on the *id* parameter, the DeviceElement subscribes for a number of parameters, the message IDs of these are as shown below:

{id}.fault: The device's summary fault state. May be one of**OK.**, **WARNING** or **Summary**

FAULT

{id}.faults.99: (Boolean) The device's communication fault

{id}.mode: The operation mode of the device. This is one of **OPERATIONAL**, **FAULT-SUPPRESSED**, **OUT-OF-SERVICE** and **MAINTENANCE**

{id}.info.signal.on: If the value reported via this messageID is **true** or **ON**, the device is considered to be actually ON-AIR. In this case, there is an additional emblem displayed at the upper right corner of the icon signalling this state.

signalBackupId: If the value reported via this messageID equals the *backupValue*, the DeviceElement considers the device to be a backup for something else.

Icon Selection

The DeviceElement shows the state of the device by switching between several variants of of the icon. The data model defines the base version of the icon which is shown when the device is in operating state without a fault. Other states are displayed by changing the icon to a variant which has a status code appended to the base name of the icon:

Icon Name	State	Example
basename.gif	Operational and OK	
basename-**C**.gif	There is no communication to the device	
basename-**F**.gif	The device shows a FAULT	
basename-**W**.gif	The device shows a WARNING	
basename-**S**.gif	The device id in fault suppressed mode	
basename-**M**.gif	The device is in maintenance mode	
basename-**O**.gif	The device is out of service	
basename-**B**.gif	The device is OK and used as backup	

In the sat-nms software there exist several sets of these icons, the is extensible by the customer. Supported image formats / file extensions are *.gif, *.jpg and *.png.

The pseudo code sniplet below describes how the DeviceElements is expected to combine the individual status variables to the name of icon to be shown.

```
if summary-alarm
    if communication-alarm
        use -C icon
    else
        use -F icon
else if summary-warning
    use -W icon
else if mode == MAINTENANCE
    use -M icon
else if mode == OUT-OF-SERVICE
    use -O icon
else if mode == FAULT-SUPPRESSED
    use -S icon
else if is-backup
    use -B icon
else
    use standard icon
```

8.5.40 DisplayElement

The DisplayElement is used to display M&C parameters read-only. It looks much alike the ParameterElement element, but never allows to change the parameter it displays.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "DisplayElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)

Key	Type	Value
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
drawFrameless	Boolean	true = display without a frame and with transparent background
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

Remarks

Conditionally Enable: If the *enableId* property is not empty, the ParameterElement subscribes for this additional parameter and disables (disabled means "dimmed" in this case) the widget unless the received parameter value matches the values stated in *enableValue*.

Font Selection: The sat-nms software defines 6 preset fonts named 'small', 'plain', 'bold', 'title', 'huge' and 'typewriter'. The *font* property contains one of these names. The selected font is used for the text in the widget, the label above the entry field always is shown with the 'plain' font, unless *font* reads 'small', in this case the label is shown with the 'small' font as well. For information about fonts and font sizes see section Fonts in the appendix of this document.

Variable Background Color: With the *variableBackgroundColor* parameter set 'true', the DisplayElement listens to the additional parameter stated in *colorId* and sets the background color of the widget according to the value received. The received value is translated to a color (and optional 'bold' printing) through the *colors* array. If the received value does not match any of the values listed in the array, the first entry in the array is used as a default / fallback.

8.5.41 EditButtonElement

The EditButtonElement places an EDIT button to the screen. It appears as a simple button with the label "EDIT" on it. A screen definition containing an EDIT button switches the screen to Edit/Take mode. This means, parameter changes are not sent immediately to the server, but the screen remembers (and marks) all changes until the operator either clicks TAKE to send all changes to the server or clicks EDIT again to revert all changes made. Also see the definition of TakeButtonElement

Data Model (extends ScreenElement)

Key	Type	Value

Key	Type	Value
type	String	Always "EditButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
privilege	Number	The privilege level required to switch to EDIT mode.

8.5.42 FaultElement

The FaultElement shows a fault flag. The faults page of the predefined device windows makes use of this display element.

The software treats device faults in a special way. In the variable list of a device driver the fault flags appear as variables called "faults.00" .. "faults.99". The software automatically produces a variable "config.faults.XX" for each fault flag "faults.XX" the driver defines. This configuration variable controls the priority of the fault.

Actually, FaultElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

8.5.43 FrameElement

The FrameElement draws a sunken 3D frame, which is intended to be used to group other elements. The 3D frame's inside area is assumes to be transparent, it does not conceal the screen elements it encloses.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "FrameElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement

8.5.44 GaugeElement

The GaugeElement shows a numeric parameter value as a horizontal bar in an entry field like frame. The gauge element is capable to adjust the scale factor for the gauge automatically from the parameter's range definition. Alternatively the scale parameters may set explicitely.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "GaugeElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. This should point to a numeric parameter to make sense.
label	String	The label shown above the strip chart
font	String	The font to be used for the label
color	String	The bar color. If thresholds are defined, this color is used while the value is inside the thresholds.
useMinMax	Boolean	If true, the min/max values for the gauge are defined in *minValue*, maxValue_. If false, the GaugeElement takes the limits from the watches parameter's range definition.
minValue	double	The value corresponding the left edge of the widget
maxValue	double	The value corresponding the right edge of the widget
useThresholds	Boolean	If true, the GaugeElement checks the value against the thresholds defined in *minThreshold*, *maxThreshold* and changes the color of the bar to *belowColor* or *above_Color* accordingly.
minThreshold	double	Values below this threshold cause the widget to display a fault
maxThreshold	double	Values above this threshold cause the widget to display a fault
belowColor	String	The bar color to be used if *useThresholds* is true and the actual value is below *minThreshold*
aboveColor	String	The bar color to be used if *useThresholds* is true and the actual value is above *maxThreshold*

Key	Type	Value
tickMode	String	One of 'NONE', 'TICKS', 'TICKS-LABELS'. 'TICKS' makes the element show ticks below the bar graph to indicate the bar position for the *minValue*, *minThreshold*, *maxThreshold* and *maxValue* values. The ticks are to be drawn outside the rectangle defining the element bounds. 'TICKS-LABELS' adds labels below the tick marks, showing the numeric values of the four limit parameters. The number of precision digits shown is taken from the subscribed value's range definition.

Restrictions for colors with *useThresholds* ###

Actually the colors to be used with *useThresholds* set are not freely definable. This is a restriction of how the sat-nms M&C software stores the screen definition, it uses a fixed set of color combinations when thresholds are enabled. These are the valid combinations:

belowColor	***color***	***aboveColor***	**sat-nms mode**
#ff0000	#00ff00	#ff0000	red - green - red
#ff0000	#ffff00	#00ff00	red - yellow - green
#00ff00	#ffff00	#ff0000	green - yellow - red

Using other color combinations than those from the table above will cause unpredictable results.

Special Behavior With Level-Adjust/Level-Control Devices

Normally the GaugeElement expects a numeric value to display in the bar graph. When used with the 'gauge' or the 'gauge2' parameter of the Level-Adjust/Level-Control devices, the GaugeElement must implement a special behavior:

The gauge/gauge2 parameters contain a comma separated string with a list of numeric and textual values. The GaugeElement must parse this string and update its display from the values parsed. The content of the gauge/gauge2 comma separated list is defined as follows:

Pos.	Name	Remark
0	val	The gauge value (numeric)
1	minLabel	The label to be shown for the min value below the graph (textual). Overwrites the label derived from the *minValue* field described above.

Pos.	Name	Remark
2	min	The value corresponding the left edge of the widget (numeric). Overwrites the value defined in the *minValue* field described above.
3	maxLabel	The label to be shown for the max value below the graph (textual). Overwrites the label derived from the *maxValue* field described above.
4	max	The value corresponding the right edge of the widget (numeric). Overwrites the value defined in the *maxValue* field described above.
5	minTLabel	The label to be shown for the minThreshold value below the graph (textual). Overwrites the label derived from the *minThreshold* field described above.
6	minThreshold	The value lower threshold value (numeric). Overwrites the value defined in the *minThreshold* field described above.
7	maxTLabel	The label to be shown for the maxThreshold value below the graph (textual). Overwrites the label derived from the *maxThreshold* field described above.
8	maxThreshold	The upper threshold value (numeric). Overwrites the value defined in the *maxThreshold* field described above.

In order to maintain compatibility to future level control devices, the GaugeElement should not use the parameter names or the driver type of the device to decide if this special behavior applies. Instead, the GaugeElement should make this decision from the data message it receives:

Contains the message a 9-element comma separated list?

If yes, the element should behave as with *useMinMax*=true and *useThresholds*=true, but with the settings contained in this list instead of the ones contained in the GaugeElement data.

Contains the message a 5-element comma separated list?

If yes, the element should behave as with *useMinMax*=true and *useThresholds*=false, but with the settings contained in this list instead of the ones contained in the GaugeElement data.

In all other cases the element should try to parse the message value to floating point value and display it with the settings given in the GaugeElement data object.

8.5.45 IconElement

The IconElement shows an arbitrary GIF/JPEG picture. If a message ID is defined with the

element properties, the image displayed will change with the parameter value addressed by the message identifier.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "IconElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	the message ID this element listens to
icon	String	the initial icon file name

When displayed the first time, the Iconelement shows the image specified in*icon*. If the message ID *id* is not empty, the IconElement dubscribes for this ID and changes the displayed icon every time a new value is received. To make use of this, the name of the icon should be structured in the following way:

```
base-name.{value-part}.{extension}
```

The value-part gets replaced by the parameter value received. For a boolean value for example, two icons 'red-lamp.false.png' and 'red-lamp.true.png'. The IconElement is configured to show 'red-lamp.false.png' initially. When the parameter addressed by *id* gets updated, the display changes according to the parameter value.

8.5.46 LatchingButtonElement

The LatchingButtonElement works much like the ParameterButtonElement described later in this document, but is specialized to show and control an enumeration parameter which knows exactly two states (e.g. on/off or true/false).

When the operator changes the state of the button by clicking it once, the latching button sends the 'other' parameter value to the device. On the other hand, if some other instance in the system changes the parameter state, the latching button recognizes this and changes the up/down state of the button accordingly.

Like the parameter button, the latching button may be labeled with text or an image. To reflect the actual state, the latching button always is configred with two text string or two image names which are shown according to the actual parameter value.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "LatchingButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens for
privilege	Number	The privilege level required to change the button state / parameter value
labelUp	String	The label shown on the button when not pressed
labelDown	String	The label shown on the button when pressed
iconUp	String	The icon shown on the button when not pressed
iconDown	String	The icon shown on the button when pressed
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used for the button label
color	String	The color used for the label
mustQuery	Boolean	Show a query popup before changing the value
queryText	String	the text to show in the query
valueUp	String	The parameter value to send when the button is going up
valueDown	String	The parameter value to send when the button is going down
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

The LatchingButtonElement never uses a button label and an icon at the same time. Either *iconUp* / *iconDown* or *labelUp* / *labelDown* contain empty strings.

The *valueUp* and *valueDown* values are not only sent when the button is pressed or released. They also are used to set the button state when a parameter value is received though *id*: If the value matches *valueDown* the button goes to 'pressed' state, in all other cases the button gets released. Labels or icons change an this moment according to the new state.

8.5.47 LockButtonElement

The LockButtonElement defines a button to lock the operation of a number of device devices that no other user can change device settings. The button is in pressed state if at least one of the devices it is assigned to is locked by some user. An operator only may release the button if he set the lock by himself of if he has a privilege level of 150 or more.

Actually, LockButtonElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

8.5.48 MCPElement

The MCPElement embeds the measurement display of a spectrum analyzer device in multi channel power measurement mode into the screen.

Actually, MCPElement is not included in the API. If the original screen definition contains instances of this type, they are skipped and not included in the ScreenDefinition reported to the UI.

8.5.49 ParameterButtonElement

The ParameterButtonElement is a button which sends a parameter value if pressed. A frequently used application for the parameter button is a RF-OFF button which sends a "tx.on=OFF" to a certain device. Beside this, a parameter button also may be programmed to play a parameter setting macro.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element sends

Key	Type	Value
privilege	Number	The privilege level required to make a button click send a value
label	String	A label shown on the button
icon	String	An icon shown on the button
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
parameterValue	String	The parameter value to send
isMacroButton	Boolean	If true, this button starts a macro instead of sending a parameter
variableBackgroundColor	Boolean	true = use a variable background color
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

The ParameterButtonElement never uses a button label and an icon at the same time. Either *icon* or *label* is an empty string.

If no *parameterValue* is set, the ParameterButtonElement sends the button label or the icon name as the parameter value when clicked

Macro Buttons

The MacroButton is a special version of the ParameterButtonElement. A *isMacroButton = true* marks a ParameterButtonElement to be a MacroButton. In this case the 'id' field contains the name of the M&C where to play the macro and the macro name in this case, separated by a period. Example `VLC0001.myMacro`

Instead of calling 'poke' when the button is clicked, a POST /api/v1/playmacro/{mnc-name}/{macro-name} call must be made by the front end (Example

`POST /api/v1/playmacro/VLC0001/myMacro` .

8.5.50 ParameterElement

The ParameterElement is the common component to display and edit most types of M&C parameters. Depending on the data type of the parameter (the data type is detected automatically) the parameter elements appears as textual / numeric entry field, choice box or as display field for read only parameters.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ParameterElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
privilege	Number	The privilege level requires to change this parameter
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
useSpinButtons	Boolean	true = use spin buttons for numeric values
spinSmallIncrement	String	Small spin increment, empty = use default
spinLargeIncrement	String	Large spin increment, empty = use default
useComboBox	Boolean	true = use a searchable combobox

Key	Type	Value

editableComboBox	Boolean	true = allow free text edit in the box
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

Remarks

Widget Variants: Depending on the type of parameter and on the*height* parameter there are different widget types associated with the ParameterElement. As the parameter type is not known at the time the screen definition is read, the decision what widget type shall be used must be done after the parameter type description has been received from the server.

Parameter Type	***height < 40***	***height >= 40***
Text	single line entry field	multi line entry field, scrollable if text does not fit into the field
Numeric	single line entry field	single line entry field
Enum/Choice	drop down or combo box	scrollable list selection
r/o Text	single line display	multi line display, scrollable if text does not fit into the field
r/o Numeric	single line display	single line display
r/o Enum/Choice	single line display	single line display

Conditionally Enable: If the *enableId* property is not empty, the ParameterElement subscribes for this additional parameter and disables (disabled means "dimmed, not changeable") the widget unless the received parameter value matches the values stated in *enableValue*.

Font Selection: The sat-nms software defines 6 preset fonts named 'small', 'plain', 'bold', 'title', 'huge' and 'typewriter'. The *font* property contains one of these names. The selected font is used for the text in the widget, the label above the entry field always is shown with the 'plain' font, unless *font* reads 'small', in this case the label is shown with the 'small' font as well. For information about fonts and font sizes see section Fonts in the appendix of this document.

Operator Query: If the property *mustQuery* reads 'true', the ParameterElement pops up a 'Do you really want...' dialog before a parameter change is sent to the server. *queryText* contains the question to be shown in this case. In the question text, placeholders may used for two values: Any occurrences of the pattern **$P** get replaced by the parameter name (message ID). Any occurrences of the pattern **$V** get replaced by the new value to set. If*queryText* is empty, the ParameterElement uses a standard text instead.

Spin Buttons: With *useSpinButtons* set 'true', the ParameterElement shows spin buttons with editable numberic values. The spin buttons allow to increment/decrement the value by clicking on them. By default, a spin button click increments or decrements the lowest significant digit shown in the entry field. With the shift key hold down, the effective increment is x10. The above applies if the *spinSmallIncrement* / *spinLargeIncrement* properties are empty. If set, these values override the defaults.

ComboBox Variants: For CHOICE parameter the ParameterElement normally uses a simple drop down box. With selection parameter with a handful of choices this easy to operate. There are however situations where the ParameterElement must handle hundreds of choices (e.g. in satellite channel lists). For this the ParameterElement provides variants of the simple drop down box.

With the *useComboBox* parameter set 'true', the ParameterElement uses a special combo box, which implements some filtering on the displayed choices. The operator may enter some text into the edit field of the combo box and only choices which (partially) contain the the entered text are offered for selection.

With the *editableComboBox* parameter set 'true', the ParameterElement permits to send any entered text as the new parameter value to the server, not only one of the valid choices for this parameter.

Variable Background Color: With the *variableBackgroundColor* parameter set 'true', the ParameterElement listens to the additional parameter stated in *colorId* and sets the background color of the widget according to the value received. The received value is translated to a color (and optional 'bold' printing) through the *colors* array. If the received value does not match any of the values listed in the array, the first entry in the array is used as a default / fallback.

8.5.51 RadioButtonElement

The RadioButtonElement is a component to display and edit CHOICE type M&C parameters as a number of radio buttons. Depending on the height of the element, radio buttons are positioned in a row or in a column (*height*<=40 means position horizontally). The radio buttons are labeled with the choices of the parameter, there are as many radio buttons as choices in the parameter range.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "RadioButtonElement".

Key	Type	Value
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to
privilege	Number	The privilege level requires to change this parameter
label	String	A label shown above the field / drop down
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query. If empty, a default query text shall be used.
drawFrame	Boolean	true = draw a gray rectangle around the radio button group.
variableBackgroundColor	Boolean	true = use a variable background color.
colorId	String	The message ID to listen for the background color
colors	Array	of ColorDefinition objects. Defines how to translate values of the *colorId* parameter to color values.

8.5.52 RectElement

The RectElement draws a rectangle with a selectable color. The element may listen to a parameter value and change the color of the rectangle according to the parameter value.

Data Model (extends ScreenElement)

Key	Type	Value

Key	Type	Value
type	String	Always "RectElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to. Is only relevant if *variableLineColor* reads *true*.
fill	Boolean	true = draw the rectangle filled, not only the outline.
color	String	The color used for the rectangle.
variableLineColor	Boolean	true = set the line/fill color from the value received from the parameter addressed by *id*.
useColorTranslation	Boolean	use the colors[] translation
colors	Array	of ColorDefinition objects. Defines how to translate values of the *id* parameter to color values if *useColorTranslation* reads *true*.

Variable Line Color

With *variableLineColor=true* the RectElement will change its line/fill color depending on the value of the parameter addressed by *id*. This way the rectangle can be used as a status indicator. The way the RectElement interprets the parameter value depends on the value of *useColorTranslation*.

If *useColorTranslation=false*, the RectElement tries to decode the line color directly from the parameter value. First it does the following tests on the parameter value (not case sensitive):

Parameter Value	Line Color And Thickness
ends with **BOLD**	The line is drawn thicker than normal
starts with **BLACK**	The line is drawn with this color
starts with **WHITE**	The line is drawn with this color

Parameter Value	Line Color And Thickness
starts with **RED**	The line is drawn with this color
starts with **BLUE**	The line is drawn with this color
starts with **GREEN**	The line is drawn with this color
starts with **YELLOW**	The line is drawn with this color
starts with **GRAY**	The line is drawn with this color
contains **TRUE**	If the message ID identifies the parameter as an alarm flag (the ID contains 'fault'), the line is drawn **RED-BOLD**, for other parameters it is drawn **GREEN-BOLD**
contains **FAULT**	The line is drawn **RED-BOLD**
starts with **ON**	The line is drawn **GREEN-BOLD**

If none of the above conditions apply, the RectElement tries to interpret the parameter value as a hex coded color (#RRGGBB). If this also fails, the standard line/fill color defined in *color* is used.

If *useColorTranslation=true*, the RectElement uses the ColorDefinition table in *colors[]* to decode the line's color and thickness.

8.5.53 SpectrumElement

The SpectrumElement embeds the spectrum display of a spectrum analyzer device in the screen. Actually, only the CSM-Spectrum-Analyzer device may be used with the SpectrumElement.

The SpectrumElement subscribes for a variety of parameter of the spectrum analyzer device to display its settings in the diagram area. It also permits to command some aspects of the spectrum analyzer device like the marker position. The spectrum data itself is read by subscribing for the 'trace' parameter of the device. This returns the spectrum data as a SpectrumTrace with each sweep of the spectrum analyzer. A complete description of the function of the spectrum element is given in a separate document.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "SpectrumElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name of the spectrum analyzer which shall be shown.
privilege	Number	The privilege level requires to change settings of the spectrum analyzer.
label	String	A label shown above the element
font	String	The font to be used for the label
color	String	The color used for the label (#RRGGBB)

8.5.54 SwitchElement

The SwitchElement is a special version of the device element which may be used to visualize the position of a switch in a user interface screen designed as a block diagram. The switch icon has all capabilities of a plain device icon display element. The context menu shown with this right mouse button additionally contains an option to toggle the switch position.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "SwitchElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device ID this element represents in the UI. The device ID consists of the M&C name followed by a dot and the device name.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.

Key	Type	Value
icon	String	The (base-) name of the icon to show
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field
touchScreenMode	Boolean	true = use a different click response, optimized for touch screen
font	String	The font to be used for the label
color	String	The color used for the label (#RRGGBB)
mustQuery	Boolean	true = show a query popup before changing the value
queryText	String	The text to show in the query

Based on the *id* parameter, the SwitchElement subscribes for a number of parameters, the message IDs of these are as shown below:

{id}.fault: The device's summary fault state. May be one of**OK.**, **WARNING** or **Summary FAULT**

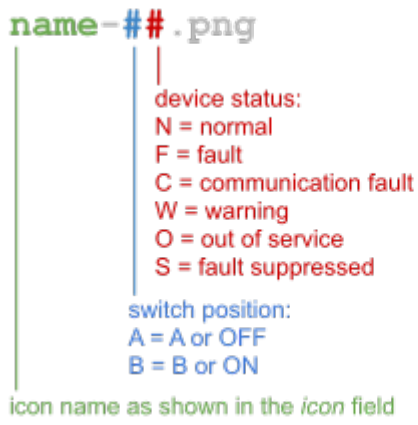
{id}.faults.99: (Boolean) The device's communication fault

{id}.mode: The operation mode of the device. This is one of**OPERATIONAL**, **FAULT-SUPPRESSED**, **OUT-OF-SERVICE** and **MAINTENANCE**

{id}.position: The switch position. May be either 'A' or 'B'. Recognized 'ON'/'OFF' es well and changes the the command sent with a switch toggle accordingly.

Icon Selection

The SwitchElement displays one icon of a given set depending on the switch position and the state of the switch device. For this the SwitchElement appends a '-' character, a 2-character status code and the suffix '.png' to the icon name given in *icon* in order to ge the file name to load / display:



8.5.55 TakeButtonElement

The TakeButtonElement places an TAKE button to the screen. It appears as a simple button with the label "TAKE" on it. The TAKE button complements the EDIT button described above in this document. It sends the changed parameter values to their destination when pressed.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TakeButtonElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement

8.5.56 TargetListElement

The TargetListElement embeds the list of targets of a SatService-ACU-ODM antenna controller in the screen. Targets may be recalled (which moves the antenna to the stored position and sets the tracking parameters associated with this target), saved or deleted.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TargetListElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement

Key	Type	Value
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The device name of the SatService-ACU-ODM. In device screens the *id* has to be interpreted differently, see below.
privilege	Number	The privilege level required to show details about the device and to change the device's operating state.
enableId	String	The enable ID (may be empty)
enableValue	String	The value which must be matched to enable the field

Device Name Indirection

The *id* parameter is interpreted differently depending on the context where the ODM Target List screen element resides:

- When placed in a user defined screen or in the main screen of the application, *id* is the name of the ODM device it shall refer to.
- When placed in the device screen of another device which defines a configuration variable with the antenna controller's device name, *id* is the name of this configuration variable and the name of the ODM device is derived from the content of this variable.
- Finally, when used in the device screen of the ODM device itself, *id* is '@'.

Target List Parsing

The target list element gets its information from a parameter 'target.list' the ODM device provides for this purpose. This variable contains the target list as a one line string with the target definitions appearing at fixed character positions.

Each target definition contains the target name and the azimuth / elevation / polarization angles. The target number is defined implicitly from the position of the target in the complete string. The format of each target definition is as follows:

```
target name (azimuth/elevation/polarization)
```

This string is padded with spaces to 45 characters length. With target numbers starting at 0, you can access a particular target in the string at position *n*45* with a length of 45 characters.

8.5.57 TextElement

The TextElement shows a label (a one line text string) with a chosable font / color.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TextElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
label	String	The text to be drawn
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)

8.5.58 ThumbnailElement

The ThumbnailElement displays a thumbnail of the video actually processed by a device like an encoder, decoder or gateway. The displayed icon is a static image, fetched directly from the device and updated every couple of seconds.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "ThumbnailElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
id	String	The message ID this element listens to

REMARKS

The Thumbnail Icon element itself must fetch the image to display in regular intervals directly from the device. As a consequence of this the following restrictions apply:

- The device to get the thumbnail images from must support this feature.
- The device to get the thumbnail images from must be accessible in the network from the client's point of view.

With *urlType = PLAIN*, the *imageUrl* is used without modification to fetch the image. The image received from the device must be scaled to fit the screen element size.

With *urlType = ADVANCED* the front end appends some parameters to the URL as shown in the example below. In this case no scaling is done as the software queries the image already in the correct size from the device.

```
http://1.2.3.4/somepath?t=1693050862003&w=320&h=180
```

The example above starts from a *imageUrl = http://1.2.3.4/somepath* definition. With *urlType = ADVANCED* the front end must add the parameters *t*, *w*, and *h* as shown. The value of *t* is the actual time in milliseconds since Jan 1st 1970, 00:00 UTC, *w* and *h* define the dimension of the image in pixels. They usually math the *width* and *height* fields of the data model.

8.5.59 XYChartElement

This element shows the relation of two numeric variables in an X/Y diagram, featuring a 'trace' which shows the recent history of the values with a configurable depth. The update rate, the diagram scaling and much more is configurable with this screen element.

Data Model (extends ScreenElement)

Key	Type	Value
type	String	Always "TextElement".
xpos	Number	see ScreenElement
ypos	Number	see ScreenElement
width	Number	see ScreenElement
height	Number	see ScreenElement
idx	String	The message ID of the parameter to show at the x-axis
idy	String	The message ID of the parameter to show at the y-axis
label	String	The label to be drawn above the diagram
font	String	The font to be used
color	String	The color used for the label (#RRGGBB)
divisions	Number	The number of divisions shown in the diagram for both directions. Typical values are 2 ('hair cross'), 4 or 10.
interval	Number	The update time interval for the display in seconds. 0.1 means to add every 100msec a new value to the display and remove the oldest value from the buffer at the same time.

Key	Type	Value
bufsize	Number	The display maintains a "first in first out" buffer of a size defined with this parameter. The buffer provides a short time memory the display shows as a trace of past values. Typical values are in the range 100-300.
xorigin	Number	The value to be shown at the center of the x axis
yorigin	Number	The value to be shown at the center of the y axis
xscale	Number	The x axis scale in 1/division
yscale	Number	The y axis scale in 1/division
limits	Boolean	If true, the display monitors the actual X/Y values to be within the limits defined below. The limit values are shown as a dark red rectangle. When the actual X/Y values exceed the limits, the diagram background becomes red.
minx	Number	The lower x limit to check. Only meaningful if *limits* is true.
maxx	Number	The upper x limit to check. Only meaningful if *limits* is true.
miny	Number	The lower y limit to check. Only meaningful if *limits* is true.
maxy	Number	The upper y limit to check. Only meaningful if *limits* is true.

8.5.60 ColorDefinition

A ColorDefinition data object defines an entry of a value to color translation table which appears as the *colors* array with several ScreenElement definitions.

Data Model

Key	Type	Value
value	String	The parameter value for which the color shall be used
color	String	The color value (#RRGGBB)
bold	Boolean	true draws the text bold if the value matches

8.5.61 ItemList

The ItemList data object is used by the API at several places to return a list of named items to the front end, e.g. when a list of existing macros or device presets is required.

Data Model

Key	Type	Value
items	Array of String	The item list

8.5.62 Macro

A Macro data object contains a macro of the M&C software, a multi-line text in the sat-nms macro language containing settings / commands to be executed.

Data Model

Key	Type	Value
name	String	The name of this macro
macro	String	The macro itself (multi-line text, lines separated by '\n' character sequences).

8.5.63 PresetValue

Contains one parameter / value definition in a DevicePreset.

Key	Type	Value
name	String	The name of the driver variable this refers to. The driver variable name is the message ID of a parameter with the M&C name and the device name stripped off.
value	String	The value to be assigned to *varName* when the preset get applied to a device.

Values appear in the same way here as in the device preset file. This is formatted along the Range definition for this driver variable:

- FLOAT parameters are formatted with the precision defined in the range, but without a trailing unit string.
- INTEGER parameters appear as simple integer values, also without unit string.
- HEX parameters appear as hexadecimal values without unit string, letters A-F are upper case.
- BOOLEAN parameters are written as TRUE/FALSE.
- CHOICE parameters show the selected choice string.
- TEXT parameters are written without quotes, this means trailing blancs are not stored (this is a general limitation of the sat-nms M&C software)

8.5.64 DevicePreset

A DevicePreset data object contains a device preset of the M&C software.

Data Model

Key	Type	Value
name	String	The name of this preset
driver	String	The name of the device driver this preset is intended to be used with.
comment	String	The comment line appearing as the first line of a device preset file in the sat-nms M&C. Does not include the leading comment character ('#') appearing in the sat-nms preset file.
preset	Array of PresetValue	The value definitions from the device preset and an array of PresetValue definitions.

8.5.65 PresetVars

A PresetVars data object contains a description of an M&C device driver with its preset-storable variable definitions.

Data Model

Key	Type	Value
driver	String	The name of the device driver.
comment	String	The content of the COMMENT statement in the driver (includes the driver version).
vars	Array of Range	The Range definitions of all driver variable which may be stored in a preset. This includes all writable variables which have neither the SETUP nor the NOPRESET property set. The list contains a Range object for each variable with the data type, limits, choices, unit and precision as defined in the device driver file. The *messageId* field of each Range object contains the local name of the variable, not a fully qualified message ID

8.5.66 DeviceVars

A DeviceVars data object contains a description of an M&C device driver with all its variable definitions.

Data Model

Key	Type	Value
driver	String	The name of the device driver.

Key	Type	Value
comment	String	The content of the COMMENT statement in the driver (includes the driver version).
vars	Array of Range	The Range definitions of all driver variables. The list contains a Range object for each variable with the data type, limits, choices, unit and precision as defined in the device driver file. The *messageId* field of each Range object contains the local name of the variable, not a fully qualified message ID

8.5.67 KeyValPair

A KeyValPair defines a key / value pair as used with the Dictionary data model.

Data Model

Key	Type	Value
key	String	The name of the key
val	String	The value assigned to *key*

8.5.68 Dictionary

The Dictionary data model contains a list of key / value pairs.

Data Model

Key	Type	Value
dict	Array	An array of KeyValPair data objects

8.5.69 SatDbSatellite

The SatDbSatellite data record contains the basic properties for one satellite in the *satellites* database table. The fields in the SatDbSatellite data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
satelliteId	Number	unique, primary key, auto generated

Key	Type	Value
name	String	free text, typically the name as given by satellite operator
operatorId	Number	relation to table *satellite-operators*
noradNumber	Number	unique name (world wide standardized), also relation to table *norad_tle*
intDesignator	String	unique name (world wide standardized)
orbitPosition	Number	satellite orbit position in degrees
inclination	Number	satellite inclination in degrees
alias	String	free text, can be used by customer to add own name
modified	String	timestamp, generated by database at creation / modification time

8.5.70 SatDbBeaconAttenuation

The SatDbBeaconAttenuation data record contains the attenuation value stored in the database for one particular pair of antennaId and beaconId. This is the attenuation to be applied for the given beacon signal when receivbed by this antenn.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
beaconId	Number	ID of the beacon this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation	Number	receiver attenuation, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

8.5.71 SatDbBeacon

The SatDbBeacon data record contains the basic properties for one satellite in the *satellite_beacons* database table. The fields in the SatDbBeacon data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There may be multiple beacon definitions assigned to one satellite. These beacons share the same *satelliteId*. One of these beacons is the default beacon, this and only this beacon has the

defaultBeacon flag set *true*.

Data Model

Key	Type	Value
beaconId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this beacon belongs to
frequency	Number	beacon frequency, floating point, MHz
polarization	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultBeacon	Boolean	true means "this is the default beacon for this satellite"
comment	String	free text describing the purpose of this beacon
modified	String	timestamp, generated by database at creation / modification time

8.5.72 SatDbTcAttenuation

The SatDbTcAttenuation data record contains the attenuation values stored in the database for one particular pair of antennaId and tcId. This is the attenuation to be applied for the given TC signal when transmitted by this antenna. The two attenuation values are applied to the same signal at different stages of the transmit chain, e.g. at the upconverter and the HPA.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
tcId	Number	ID of the TC signal this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation1	Number	transmit attenuation1, floating point, dB
attenuation2	Number	transmit attenuation2, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

8.5.73 SatDbTc

The SatDbTc data record contains the basic properties for one satellite in the*satellite_tc* database table. The fields in the SatDbTc data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how

relations between tables are maintained using IDs.

There may be multiple telecommand channel definitions assigned to one satellite. These channels share the same *satelliteId*. One of these channels is the default TC channel, this and only this channels has the *defaultTc* flag set *true*.

Data Model

Key	Type	Value
tcId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this TC channel belongs to
frequency	Number	transmit frequency, floating point, MHz
polarization	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultTc	Boolean	true means "this is the default TC channel for this satellite"
comment	String	free text describing the purpose of this channel
modified	String	timestamp, generated by database at creation / modification time

8.5.74 SatDbTmAttenuation

The SatDbTmAttenuation data record contains the attenuation values stored in the database for one particular pair of antennaId and tmId. These are the attenuation values to be applied for the given primary and secondary TM signals when received by this antenna.

Data Model

Key	Type	Value
attenuationId	Number	unique, primary key, auto generated
tmId	Number	ID of the TM signal this data record belongs to
antennaId	Number	ID of the antenna this beacon belongs to
attenuation1	Number	transmit attenuation1, floating point, dB
attenuation2	Number	transmit attenuation2, floating point, dB
modified	String	timestamp, generated by database at creation / modification time

8.5.75 SatDbTm

The SatDbTm data record contains the basic properties for one satellite in the*satellite_tm* database table. The fields in the SatDbBeacon data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There may be multiple telemetry channel definitions assigned to one satellite. These channels share the same *satelliteId*. One of these TM channels is the default channel, this and only this channel has the *defaultTm* flag set *true*.

Data Model

Key	Type	Value
tmId	Number	unique, primary key, auto generated
satelliteId	Number	ID of the satellite this beacon belongs to
frequency1	Number	primary receive frequency, floating point, MHz
frequency2	Number	secondary receive frequency, floating point, MHz
polarization1	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
polarization2	String	this is an enumeration, may be one of "H", "V", "RHCP" or "LHCP"
defaultTm	Boolean	true means "this is the default telemetry channel for this satellite"
comment	String	free text describing the purpose of this beacon
modified	String	timestamp, generated by database at creation / modification time

8.5.76 SatDbSatOperator

The SatDbSatOperator data record contains the basic properties for one satellite in the *satellite_operators* database table. The fields in the SatDbSatOperator data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

There is a limited list of satellite operators, each being assigned to multiple satellites. The relation is made by the *operatorId* which is stored with each satellite.

Data Model

Key	Type	Value

Key	Type	Value
operatorId	Number	unique, primary key, auto generated
name	String	free text, company name of satellite operator
contact	String	free text, multiline, line breaks are coded as '\n' character sequences.
modified	String	timestamp, generated by database at creation / modification time

8.5.77 SatDbPosition

The SatDbPosition data record contains the antenna pointing information for one satellite and one antenna in the *satellite_positions* database table. The fields in the SatDbPosition data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

The *Satellite-Positions* table typically contains *number-of-satellites* x *number-of-antennas* records, as each satellite requires different antenna pointing angles for each particular antenna.

Data Model

Key	Type	Value
positionId	Number	unique, primary key, auto generated
satelliteId	Number	the ID of the satellite this SatDbPointing record refers to (relates to the *satellites* table).
azimuth	Number	antenna azimuth pointing angle in degrees (floting point)
elevation	Number	antenna elevation pointing angle in degrees (floting point)
polarization	Number	antenna polarization pointing angle in degrees (floting point)
satelliteColocation	Array of Number	contains the list of satellites (referenced by their satelliteId) which are seen at the same pointing angles by this particular antenna. See remark below
targetNumber	Number	a target number to be set at some antenna controllers (integer)
antennaId	Number	the ID of the antenna this SatDbPointing record refers to (relates to the *antennas* table).

Key	Type	Value
defaultPosition	Boolean	true means "this is the default position for this antenna"
comment	String	a free text describing the purpose of this database entry
modified	String	timestamp, generated by database at creation / modification time

Remark: Depending on the dish size of an antenna, the antenna "sees" not only the satellite it is pointed to but also other satellites which are at positions in the orbit very near to the first satellite. The *satelliteColocation* list contains the satelliteIds for these satellites.

Actually this list is not maintined by the the backend, the *satelliteColocation* will always read as an empty array.

8.5.78 SatDbAntenna

The SatDbAntenna data record describes one antenna in *antennas* database table. The fields in the SatDbPosition data record are 1:1 replicas of the SQL database record. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
antennaId	Number	the ID of the antenna this SatDbAntenna record refers to.
devicename	String	the name of the sat-nms device. controlling this antenna. The name has then name of the M&C system prepended, e.c. "VLC0001.ANTENNA-1"
alias	String	a free text describing the antenna.
modified	String	timestamp, generated by database at creation / modification time

8.5.79 SatDbTLEData

The SatDbTLEData data record contains the TLE ephemeris data for one satellite. The fields in the SatDbTLEData data record are 1:1 replicas of the SQL database record from the *norad_tle* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
noradNumber	Number	the norad number of the satellite this SatDbTLEData record refers to. As the norad number is unique, this field is used as the primary key in the database table.
tleName	String	the name of the satellite as stated in the original TLE parameter set.
tleLine1	String	the first line of TLE data
tleLine2	String	the second line of TLE data
modified	String	timestamp, generated by database at creation / modification time

Remark: The TLE (aka Nasa Two Line Elements) format identifies single parameters by their exact position in the lines 1/2. This requires that the *tleLine1* and *tleLine2* fields are processed exactly as they are, there must no whitespace be added or removed from the start or end of the value, multiple blanc characters must not be compressed to one. This is essential to keep the values readable.

8.5.80 SatDbI11Data

The SatDbI11Data data record contains the Intelsat 11 parameter ephemeris data for one satellite. The fields in the SatDbI11Data data record are 1:1 replicas of the SQL database record from the *intelsat_11parameters* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
i11Id	Number	a unique ID of this SatDbI11Data record (auto generated).
satelliteId	Number	the ID of the satellite this SatDbI11Data record refers to.
i11Name	String	the name of the satellite as stated in the original I11 parameter set.
i11Data	String	the 11 ephemeris parameters, floating point, separated by semicolon characters
defaultI11	Boolean	true means "this is the default I11 dataset for this satellite"
comment	String	free text describing the purpose of this dataset
modified	String	timestamp, generated by database at creation / modification time

8.5.81 SatDbTableTracking

The SatDbTableTracking data record contains the table tracking information for one satellite. The fields in the SatDbTableTracking data record are 1:1 replicas of the SQL database record from the *table_tracking* table. Please refer to the database model description at section Satellite Database how relations between tables are maintained using IDs.

Data Model

Key	Type	Value
ttId	Number	a unique ID of this SatDbTableTracking record (auto generated).
satelliteId	Number	the ID of the satellite this SatDbTableTracking record refers to.
antennaId	Number	the ID of the antenna this SatDbTableTracking record refers to.
filename	String	the name of the data file containing the position data of the satellite. Should be treated as a free text entry field in the UI.
defaultTt	Boolean	true means "this is the default table tracking dataset for this satellite"
comment	String	free text describing the purpose of this dataset
modified	String	timestamp, generated by database at creation / modification time

8.5.82 SatDbSatDetails

The SatDbSatDetails data record contains detailed information for one satellite. It is returned by a GET performed on the /api/v1/satdetails endpoint, collecting all information known about the satellite in one single document.

Data Model

Key	Type	Value
satellite	SatDbSatellite	the basic properties of the satellite
operator	SatDbSatOperator	the satellite operator assigned to this satellite
beacons	Array of SatDbBeacon	a list of all beacons defined for this satellite, may contain 0 or more elements.
tcChannels	Array of SatDbTc	a list of all telecommand channels defined for this satellite, may contain 0 or more elements.
tmChannels	Array of SatDbTm	a list of all telemetry channels defined for this satellite, may contain 0 or more elements.

Key	Type	Value
tleData	Array of SatDbTLEData	the TLE ephemeris data defined for this satellite, may contain 0 or 1 elements.
i11Data	Array of SatDbI11Data	the I11 ephemeris data defined for this satellite, may contain 0 or more elements.
tableData	Array of SatDbTableTracking	the table tracking information for this satellite, may contain 0 or more elements.

The SatDbSatDetails data record does not contain any antenna pointing information as this also depends on the antenna used to see this satellite.

8.5.83 SatDbTleImport

The SatDbTleUpload data record is used to upload a file containing a list of TLE definitions to the satellite database. It is used with the /api/v1/tleimport API endpoint.

Data Model

Key	Type	Value
fileName	String	the name of the file which is to be imported. This is used in the backend for logging etc.
replaceAll	Boolean	Controls the behavior of the import process, FALSE merges the database and file content together, overwiting data sets with the same norad number. TRUE ensures that after the import only the TLE datasets from imported file are in the database.
fileContent	String	The content of the file to be imported, BASE64 coded.

8.5.84 SatDbState

The SatDbState data record reports the state of the connection of the backend to the database server. The state may be queried by a GET /api/v1/dbstate API call.

Data Model

Key	Type	Value
connected	Boolean	true if the backend is connected to a DB server
primary	Boolean	true if the backend is connected to the primary DB server
serverAddress	String	the IP address of the DB server the backend is actually connected to

8.5.85 ChannelData

The ChannelData data record holds the information about one satellite channel stored in the satellite channel database. It is used by the /api/v1/satellites/##/channels API calls

Data Model

Key	Type	Value
channelId	Number	A unique ID of this ChannelData record (auto generated).
satelliteId	Number	The ID of the satellite this ChannelData record refers to.
name	String	The user defined channel name.
fullName	String	The full name of this channel as it appears as a selection in the user interface. The full name is built from the satellite's name, a '-' character and the *name* field in this record. The full name must be unique because the operator selects satellite channels by this name. With POST or PATCH calls to the channel database the backend ignores this field, it automatically generates the *fullName* from the satellite name and *name* .
user	String	The name of the user who lastly edited the channel
programTitle	String	Name of the program transmitted or received with this channel
comment	String	A one line comment describing this satellite channel
rxFrequency	Number	The receive frequency in MHz (3 digits precision)
rxPolarization	String	The receive polarization, one of 'X', 'Y', 'L' or 'R'
txFrequency	Number	The transmit frequency in MHz (3 digits precision)
txPolarization	String	The transmit polarization, one of 'X', 'Y', 'L' or 'R'
dvbMode	String	The DVB mode, one of 'AMCD', 'AMCDL', 'AMCDVBS', 'AMCNBC', 'AMDVBS', 'ATSC', 'AUTO', 'DSNG', 'DSS', 'DVBS', 'DVBS2', 'DVBS2X', 'DVBT', 'DVBT2', 'NS3', 'NS4' or 'TURBO'
nlcMode	String	The NLC mode, one of 'OFF' or 'ON'
symbolRate	Number	The symbol rate in Msym/sec (4 digits precision)

Key	Type	Value
fec	String	The FEC rate, one of 'AUTO', '1/1', '1/16', '1/2', '1/3', '1/32', '1/4', '1/5', '1/8', '2/2', '2/3', '2/5', '2/9', '3/16', '3/2', '3/3', '3/4', '3/5', '4/15', '4/2', '4/3', '4/5', '4/9', '5/16', '5/2', '5/6', '5/9', '6/2', '6/5', '6/7', '7/15', '7/2', '7/8', '7/9', '8/10', '8/15', '8/16', '8/2', '8/9', '9/10', '9/2', '9/20', '10/11', '10/2', '11/12', '11/15', '11/2', '11/20', '11/45', '12/2', '12/24', '13/14', '13/15', '13/18', '13/2', '13/24', '13/30', '13/45', '14/24', '14/45', '15/16', '15/25', '16/25', '17/30', '19/20', '19/30', '21/22', '21/44', '22/45', '23/36', '25/36', '26/45', '28/45', '29/45', '31/45', '32/45', '37/45', '41/5', '43/27', '44/27', '45/180', '45/28', '46/28', '47/26', '48/26', '50/15', '55/56', '60/180', '64/57', '70/140', '70/150', '72/180', '77/90', '80/100', '80/180', '90/180', '90/30', '100/180', '108/180', '114/180', '120/180', '126/112', '126/180', '135/180', '144/180', '150/180', '160/180', '162/180', '188/204', '194/178', '204/188', '208/192', '219/201', '225/205', '380/400', '442/485', '485/422', '510/511' or '510/512'
modulation	String	The modulation type, one of '128APSK', '128QAM', '16APSK', '16PSK', '16QAM', '16QUAM', '256APSK', '256QAM', '32APSK', '32QAM', '64APSK', '64QAM', '8APSK', '8PSK', '8QAM', 'AQPSK', 'AUTO', 'BPSK', 'OQPSK', 'PSK8', 'QAM', 'QAMAUTO', 'QPSK', 'SEQ', 'SOQPSK', 'SQPSK', 'UQPSK', 'VIT', 'VSB16' or 'VSB8'
rolloff	String	The roll off factor, one of '0.02', '0.05', '0.10', '0.15', '0.20', '0.25' or '0.35'
pilots	String	The pilots on/off switch, one of 'OFF' or 'ON'
bitRate	Number	The data rate in Mbit/sec (4 digits precision)
useRate	String	Selects the way the data rate is specified, one of 'SYMBOLRATE' or 'BITRATE'
packetSize	String	The packet size, one of '188' or '204'
frameSize	String	The frame size, one of 'SHORT' or 'NORMAL'
videoRate	Number	The video data rate in Mbit/sec (4 digits precision)
vrAuto	String	The video rate auto switch, one of 'FIXED' or 'AUTO'

Key	Type	Value
profile	String	The video encoding profile, one of 'MPEG2-MP@ML-420', 'MPEG2-MP@LL-420', 'MPEG2-HP@ML-422', 'MPEG2-HP@LL-422', 'MPEG2-SP@LL-420', 'MPEG2-SP@ML-420', 'MPEG2-MP@H14-420', 'MPEG2-MP@HL-420', 'MPEG2-HP@H14-422', 'MPEG2-HP@HL-422', 'MPEG2-HP@HL-420', 'MPEG2-HP@H14-420', 'MPEG2-HP@ML-420', 'MPEG4-BP@L2-420', 'MPEG4-BP@L3-420', 'MPEG4-BP@L4-420', 'MPEG4-MP@L3-420', 'MPEG4-MP@L4-420', 'MPEG4-MP@L4.2-420', 'MPEG4-HP@L3-420', 'MPEG4-HP@L4-420', 'MPEG4-HP@L4.2-420', 'MPEG4-HP@L3-422', 'MPEG4-HP@L4-422' or 'MPEG4-HP@L4.2-422'
eirp	Number	The transmit EIRP in dBW (2 digits precision)
redEirp	Number	The reduced EIRP for line up in dBW (2 digits precision)
autoLineup	String	Configures automatic line up, one of 'OFF' or 'ON'
audioRate1	String	Audio channel 1 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate2	String	Audio channel 2 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate3	String	Audio channel 3 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioRate4	String	Audio channel 4 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audioChannels	Number	Number of audio channels, (integer, 1-4)
decoderInput	String	Decoder input, one of 'SAT', 'IP' or 'ASI'
ipType	String	IP adressing type, one of 'UNICAST' or 'MULTICAST'
ipProtocol	String	IP protocol, one of 'UDP', 'RTP', 'TCP' or 'ZIXI'
ipSourceAddress	String	Source IP address in dotted quad notation
ipSourcePort	Number	Source port number (integer, 0-65535)
ipDestinationAddress	String	Destination IP address in dotted quad notation

Key	Type	Value
ipDestinationPort	Number	Destination port number (integer, 0-65535)
ipFec	String	IP FEC selection, one of 'OFF' or 'ON'
ipBufferSize	Number	Buffer size (integer)
encoderPhysInterface	Number	Encoder physical output number (integer, 1-4)
decoderPhysInterface	Number	Decoder physical input number (integer, 1-4)
modified	String	timestamp, generated by database at creation / modification time

8.5.86 StreamKeyData

The StreamKeyData data record holds the information about one BISS or SRT key in the *streamkeys* table of the database . It is used by the /api/v1/streamkeys API calls.

Data Model

Key	Type	Value
keyId	Number	A unique ID of this StreamKeyData record (auto generated).
name	String	The user defined name of this stream key. This name also must be unique because the operator selects the key by this name.
keyType	String	The key type, one of 'BISS-1', 'BISS-E' or 'PASSPHRASE'.
key	String	The key value associated with the id and name in this record. For biss keys this is a string entirely consisting of the hexadeximal characters 0123456789ABCDEF, it is either 12 or 16 characters long. For other key types the key is free text.
modified	String	timestamp, generated by database at creation / modification time

8.5.87 TreeViewNode

The TreeViewNode data record holds the information about one node in the sat-nms tree view and the tree below this node. The complete tree is partially autogenerated and user defined, see below about the standard tree structure used in the sat-nms software.

Data Model

Key	Type	Value

Key	Type	Value
nodeName	String	The (displayed) name of this node. This may be a M&C name, a subsystem name, a device name or a screen name, depending on the type of this node. See below on details about node types and names.
nodeType	String	One of ROOT, MNCNODE, DEVICELIST, SUBSYSTEM or DEVICE. See below on details about node types and names.
children	Array	An array of other TreeViewNode objects being the children of this node. May be null, this denotes a leaf node.

Standard Tree Structure

The general structure of the tree view ist autogenerated by the software. User defined branches with subsystem definitions are integrated at certain points of the tree.

```
ROOT
  +-- MNC001
  +-- MNC002
     +-- Devices
     +-- SUBSYSTEM-1
     +-- SUBSYSTEM-2
        +-- SUBSUBSYSTEM-1
           +-- DEVICE-1
           +-- DEVICE-2
     +-- DEVICE-3
     +-- DEVICE-4
  +-- MNC003
```

A tree always consists of one ROOT node. The children of this node are the M&C systems managed by the backend (one MNCNODE for each M&C system). Each MNCNODE has at least one child, a DEVICELIST node containing the list of devices managed by this M&C. Up to here all nodes are auto-generated and not c hangeable by the user. Additionally, each MNCNODE may have zero or more SUBSYSTEM nodes with user defined groups of devices or nested subsystems.

Node Types and Names

A TreeViewNode may be one of the types ROOT, MNCNODE, DEVICELIST, SUBSYSTEM or DEVICE. The table below lists the properties of these node types in short form:

type	***auto-generated***	***may contain***	***name***
ROOT	yes	MNCNODE	always 'TreeView'

type	*auto-generated*	*may contain*	*name*
MNCNODE	yes	DEVICELIST, SUBSYSTEM	M&C name
DEVICELIST	yes	DEVICE	always 'Devices'
SUBSYSTEM	no	SUBSYSTEM, DEVICE	subsystem name
DEVICE	within a DEVICELIST	-/- (leaf node)	device name

The **ROOT** node is - as the name suggests - the root of the tree view tree. There is exactly one **ROOT** node in the tree, its children are one or more**MNCNODE** nodes, as many as M&C systems are configured to be managed by the backend. The displayed name of each **MNCNODE** node id the ID / name of this M&C, this is the same as used in message IDs. The list of **MNCNODE** nodes is auto-generated by the backend, derived from its configuration data.

The first child of each **MNCNODE** node always is a **DEVICELIST** node containing the complete list of devices configured on this M&C. This node also is automatically created by the backend, the device list is in 'natural' order, meaning that the devices appear in the same order as they do in the M&C configuration file / screen.

DEVICE nodes are the leaf nodes of the tree. The name of a**DEVICE** node is the name of the device without the name of the M&C it is contained in.

SUBSYSTEM nodes define a group of devices or nested subsystems. They are the only user defined node objects in the tree. The name of a **SUBSYSTEM** node is user defined, but as this name appears in the message ID of the summary fault state of a subsystem, the same restrictions applay for subsystem names as for device names. They may consist of upper case letters 'A' .. 'Z', digits and the '-' - character.

Subsystems

Subsystems defined by a **SUBSYSTEM** node in the tree are managed by the M&C like a virtual device which has a summary fault flag which reads as the maximum fault condition that appears at one of the elements the subsystem contains. The definition of all configured subsystems in a M&C is stored by the M&C itself, not by the backend. Due to the M&C-local architecture of subsystems the following rules apply to them:

The name of a subsystem consists of the same limited set of characters like device names do.

A subsystem may contain devices or other subsystems. These children must be located at the same M&C as their parent subsystem. Subsystems cannot be defined across multiple M&Cs.

Each subsystem distributes a summary fault variable which can be used to display the fault state of the subsystem. The full messgeId of this variable is

`{M&C-name}.SUBSYSTEM.{subsystem-name}.fault`

or for nested subsystems

`{M&C-name}.SUBSYSTEM.{subsystem-name}.{sub-subsystem-name}....fault`

These variables - like a .fault value of a device - read one of the strings "OK.", "WARNING" or "Summary FAULT".

8.5.88 DatabaseVersion

The DatabaseVersion reports the actual version (migration_step) of the database together with the version required by the backend and a fault flag indicating that the database version is too old for this backend.

Data Model

Key	Type	Value
reported	Number	The version / migration_step reported by the database. a value of -1 means 'undefined'
required	Number	The version / migration_step required by the backend
fault	Boolean	True of the database version is too old

8.5.89 DebugMessage

The DebugMessage data record is used for the communication with a M&C debug terminal. It transports the lines to be displayed in the terminal via the websocket and also the commands to be sent to the debug terminal using the /api/v1/debug API call the other way round.

Data Model

Key	Type	Value
msg	String	The line to be transferred (without any newline character at the end)
error	Boolean	True if this is an error message issued by the backend rather than a message from the M&C

8.5.90 RedundancyState

The RedundancyState data record carries the actual state of a particular M&C redundancy switching logic. It is returned by a GET /api/v1/redundancy API call.

Data Model

Key	Type	Value

Key	Type	Value
mncName	String	The name of the redundant M&C pair for which this data record reports the state
enabled	Boolean	Reports if this redundancy is enabled (true/false)
state	String	Reports the actual state of the redudancy switching logic, one of 'PRIMARY', 'BACKUP', 'NONE'

8.5.91 RedundancySummary

The RedundancySummary data record carries the actual state of the M&C redundancy switching logic for all redundant M&Cs. It is returned by a GET /api/v1/redundancy API call.

Data Model

Key	Type	Value
redundantMnCs	Array	An array of RedundancyState objects, reporting the states to all redundant M&Cs. The array may be empty if no redundant M&Cs are configured in the backend.

8.5.92 RedundancyCmd

The RedundancyCmd data record carries the command to be executed with a /api/v1/redundancy API call.

Data Model

Key	Type	Value
cmd	String	The command to be executed. Valid commands are: 'ENABLE' = enable redundancy switching in case of a communication fault. 'DISABLE' = disable redundancy switching. 'RESET' = switch back to the primary IP address for the M&C 'FORCE' = force a redundancy switch to the backup IP address for the M&C even if the communication at the primary IP address works fine.

8.5.93 RestartCmd

The RestartCmd data record carries the command to be executed with a/api/v1/restart API call.

Data Model

Key	Type	Value

Key	Type	Value
cmd	String	The command to be executed. The only valid command is "RESTART". The restart is only executed if *cmd* has this value.

8.5.94 ScheduleEvent

The ScheduleEvent data record carries the data of one event in the M&C macro scheduler.

Data Model

Key	Type	Value
eventId	Number	An integer value in the range 1 .. 2147483647 uniquely identifying this event in the schedule. A value of 0 means undefined.
groupId	Number	The event's group identifyer. this is the *eventId* if the master event in the group. Even this master event has its own *eventId* set as *groupId*, a *groupId* of 0 means this event is not grouped.
state	String	One of the following: DONE, UPCOMING or MISSED
repeatMode	String	One of the following: NONE, DAILY, WEEKLY or MONTHLY
enabled	Boolean	*true* enables the execution of the event's macro, *false* inhibits this.
description	String	A description for this event given by the operator.
firstExecution	String	The time of the first execution of the event.
nextExecution	String	The time of the next execution of the event.
lastExecution	String	The date of the last execution of the event if a repeat mode is set. For non-repeating events this reads null. The time stamp should contain a time (e.g. 00:00:00) although the scheduler ignores this time.
macro	String	The macro to be executed. newline characters on the macro thave to written as the character sequence "\n".
dict	Array	An optional array of KeyValPair definitions which may be used to append additional information to the event.

Remarks

- New events shall use the lowest *eventId* above 1000 which is not used in the schedule.
- All time stamps are formatted as 'YYYY-MM-DDTHH:MM:SSZ' ans expressed in UTC.

- In the macro text newline characters are expressed as a "\n" character sequence.
- The fields *groupId* and *dict* are not used by the macro scheduler appliaction. The Web GUI shall not show these fields snd shall not present them for editing.

8.5.95 Schedule

The Schedule data record carries the complete shedule of an M&C macro scheduler. It is used with the /api/v1/schedule API call.

Data Model

Key	Type	Value
events	Array	An array of ScheduleEvent data objects defining the complete schedule.

8.5.96 DeviceDefinition

The DeviceDefinition data record carriers the information about one device in an M&C's device setup.

Key	Type	Value
name	String	The name of the device
driver	String	The device driver

8.5.97 DeviceThreadDefinition

The DeviceThreadDefinition data record carriers the information about one device thread in an M&C's device setup.

Key	Type	Value
port	String	The communication port / interface this thread uses
protocol	String	The protocol this thread uses
idle	Number	The idle time of this thread (msecs)
devices	Array	An array of DeviceDefinition objects containing the devices of this thread

8.5.98 DeviceSetup

The DeviceSetup data record carriers the complete information of an M&C's device setup.

Key	Type	Value
comment	String	A short one line description (e.g. time stamp an user name of the last change)
threads	Array	An array of DeviceThreadDefinition objects containing the device thread definitions of this setup

8.5.99 FRViewProperties

The FRViewProperties data record contains the setting stored for a file recorder viewer.

Key	Type	Value
title	String	The diagram title
traces	Array	An array of FRViewTraceDescription objects containing the trace names and unit strings. The File-Recorder device records four traces of data, so this array always contains four elements. When sent to the backend also four elements must be supplied.
presets	Array	An array of FRViewTracePreset objects containing the scaling presets defined for this file recorder view. This array always contains eight elements, when sent to the backend also eight elements must be supplied. In the Java-UI the presets are identified by position in this array, so the front end should not re-order them.

8.5.100 FRViewTraceDescription

The FRViewTraceDescription data record contains a basic description (label, unit) for one trace within the FRViewProperties data record.

Key	Type	Value
unit	String	The unit string to be displayed at the axis for this trace
name	String	The name of the trace

8.5.101 FRViewPreset

The FRViewPreset data record contains the data stored for one user preset within the FRViewProperties data record. This includes the diagram scaling and some additional information like trace colors etc.

Key	Type	Value

Key	Type	Value
absstart	String	The (absolute) starttime of the diagram, expressed as an ISO 8601 string, example: `2021-03-26T08:02:30Z` , time zone is always UTC.
relstart	Number	The starttime of the diagram, in seconds relative to the newest data record recorded.
tscale	String	The time axis scaling, one of ("15s", "30s", "1m", "3m", "5m", "10m", "30m", "1h", "3h", "6h", "12h", "1d", "2d", "6d", "30d"). The values are per division of the time axis
traces	Array	An array of FRViewTraceScaling objects, each describing the scaling properties for one diagram trace. The backend will always report four traces, even if not all four traces are configured in the File-Recorder device. When sent to the backend, each preset **must** provide four elements in this array to be recognized properly.
name	String	The (optional) name of the preset

8.5.102 FRViewTraceScaling

The FRViewTraceScaling data record contains the scaling information for one diagram trace within the FRViewPreset data record.

Key	Type	Value
yref	Number	The reference value (the value shown at the half diagram height)
yscale	Number	The Y-scale value (y-units / division with 10 divisions diagram height). The values must match a 1/2/5 stepping, they will be read back rounded to this if other values are supplied.
enabled	Boolean	TRUE means this trace shall be shown
color	String	The trace color, expressed as a hexadecimal RGB value preceeded by a '#' character

8.5.103 AcuTarget

The AcuTarget data model contains the basic properties (name, az, el, pol) for one target / satellite stored on an ACU device

Key	Type	Value
targetNo	Number	The target number on the ACU

Key	Type	Value
satelliteName	String	The name of the target
azimuth	Number	The stored azimuth angle
elevation	Number	The stored elevation angle
polarization	Number	The stored polarization angle

8.5.104 AcuTargetList

The AcuTargetList data model contains the target list read from an ACU device, showing all used target memories as an array of AcuTarget objects.

Key	Type	Value
acuName	String	Name of the ACU device
acuTargets	Array	Target list as an array of AcuTarget objects.

8.5.105 InventoryItem

The InventoryItem data record contains the data stored for one particular item in the inventory database. It is used to report, add or modify the properties of an inventory item thru the API. Beside this, the InventoryItem record is used to specify a filter when getting a list of InventoryItem records. in such a case, all fields which are missing or null are interpreted as a wildcard, specified fields must match to include the item in the list.

Key	Type	Value
itemId	Number	A unique ID of this record (auto generated).
serialNo	String	The serial number of the item / device
vendor	String	The vendor name (for serach filters)
model	String	The modem name (for serach filters)
state	String	The administrative state of the device
comment	String	free text
modified	TimeStamp	automatically created timestamp of last change

8.5.106 InventoryDevice

The InventoryDevice data record contains the device mappings for aInventoryItem. There may be multiple devices mapped to one item in the inventory database.

Key	Type	Value
deviceId	Number	A unique ID of this record (auto generated).
itemId	Number	The ID of the InventoryItem this record refers to
mncName	String	The serial number of the item / device
deviceName	String	The vendor name (for serach filters)
state	String	The administrative state of the device
online	Boolean	true means the device is present
modified	TimeStamp	automatically created timestamp of last change

8.5.107 InventoryLogEntry

The InventoryLogEntry data record contains the data of one entry in the inventory life cycle log. It is used to report, add or modify the properties of an inventory log entry the API. Beside this, the InventoryLogEntry record is used to specify a filter when getting a list of InventoryLogEntry records. in such a case, all fields which are missing or null are interpreted as a wildcard, specified fields must match to include the item in the list.

Key	Type	Value
entryId	Number	A unique ID of this record (auto generated).
itemId	Number	Links to the inventory_items entry this log entry refers to.
mncName	String	The M&C which issued this message or null if the message was entered at the UI
devName	String	The satnms device name which issued this message
message	String	the message text
created	TimeStamp	timestamp, generated by database at creation time

8.5.108 DocumentItem

The DocumentItem data record contains the data of one entry in the backend's document list. It is used to report, add or modify the properties of a stored document thru the API. Beside this, the InventoryLogEntry record is used to specify a filter when getting a list of DocumentItem records. in such a case, all fields which are missing or null are interpreted as a wildcard,

specified fields must match to include the item in the list.

Key	Type	Value
documentId	Number	A unique ID of this record (auto generated).
fileName	String	Name of the file. The name must be unique, all files are stored in the same directory.
contentType	String	Content type of the file. must be one of application/pdf, image/png or image/jpg.
fileContent	String	The file content as a BASE64 coded string
driver	String	The satnms driver name this document refers to or null if it does not refer to a driver / device type.
mncName	String	The M&C name this document refers to or null if it does not refer to a M&C.
comment	String	free text
modified	TimeStamp	automatically created timestamp of last change

8.5.109 DocumentList

The DocumentList data record contain a list of DocumentItem records.

Key	Type	Value
documents	Array	An array of DocumentItem records.

8.5.110 Thumbnail

The Thumbnail data record contains the data fro a thumbnail image when fetched using the /api/v1/thumbnail API call.

Key	Type	Value
messageId	String	The full message ID of the thumbnail image
image	String	The binary thumbnail image (jpf or png), base64 encoded.

8.5.111 BackendInfo

Key	Type	Value

Key	Type	Value
backendName	String	The current backend name
role	String	Role of current backend
isPrimary	Boolean	Whether the backend is main or not
backendNames	String []	Array of backend names (string) to set the activeBackend name

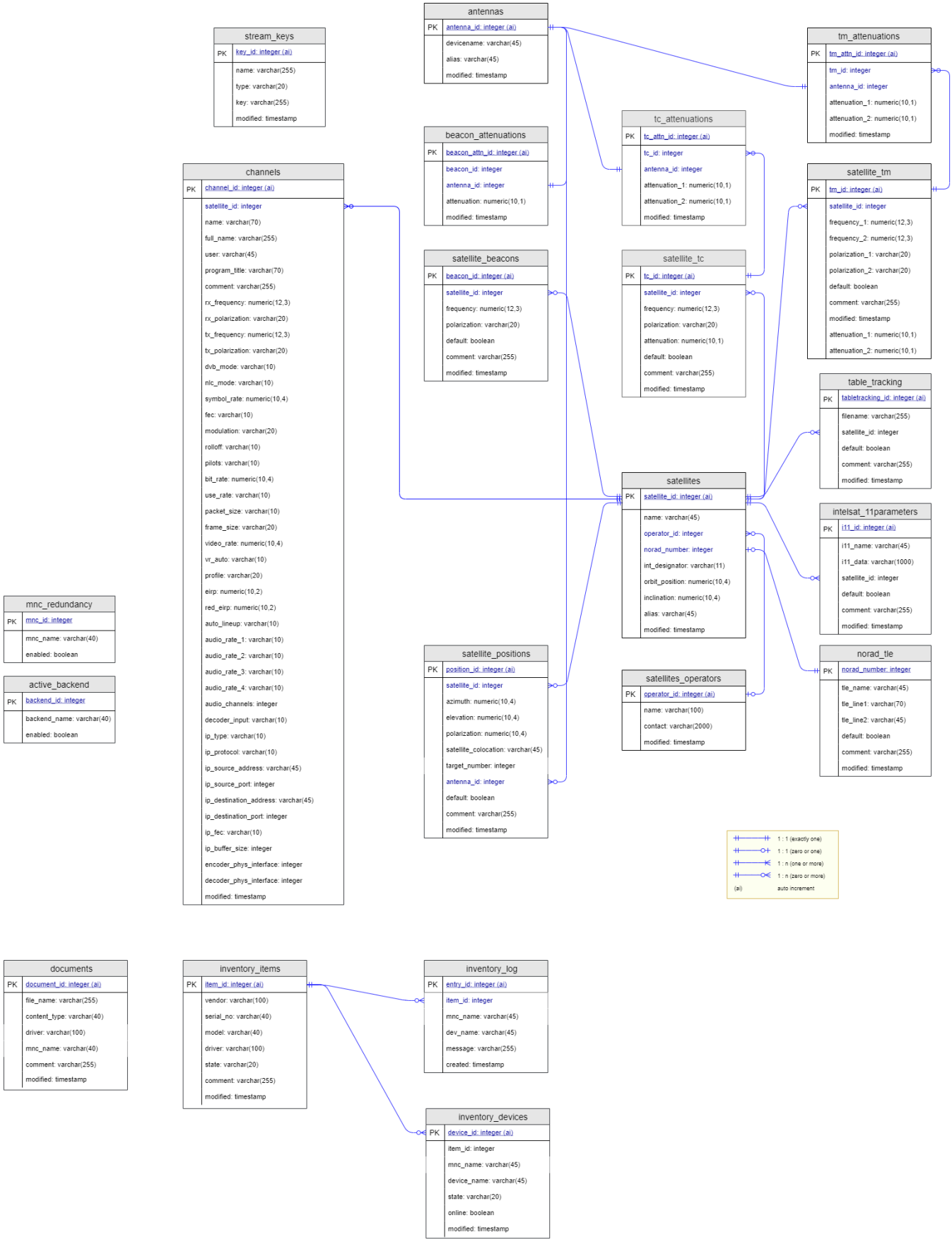
8.5.112 UISettings

The User-Settings data record contains changed settings by the user. Not saved or changed settings will be generated from Backend.

Key	Type	Value
settingsId	Integer	A Unique Id of an setting
key	String	Setting name
value	TEXT	The saved value of the setting
username	String	Username to which the setting belongs
isLocked	Boolean	Whether the setting is locked for user to overwrite it
modified	TimeStamp	automatically created timestamp of last change

8.6 Satellite Database

The satnms4 backend stores all information about satellites, antenna pointing, ephemeris data etc. In a SQL database which is accessible through the backend's API. The following paragraphs describe the table structure of the database and the meaning of each particular table column as an appendix to the API description given above.


stream_keys
PK key_id: integer (ai)
name: varchar(255)
type: varchar(20)
key: varchar(255)
modified: timestamp
antennas
PK antenna_id: integer (ai)
devicename: varchar(45)
alias: varchar(45)
modified: timestamp
tm_attenuations
PK tm_attn_id: integer (ai)
tm_id: integer
antenna_id: integer
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
modified: timestamp
tc_attenuations
PK tc_attn_id: integer (ai)
tc_id: integer
antenna_id: integer
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
modified: timestamp
beacon_attenuations
PK beacon_attn_id: integer (ai)
beacon_id: integer
antenna_id: integer
attenuation: numeric(10,1)
modified: timestamp
satellite_tm
PK tm_id: integer (ai)
satellite_id: integer
frequency_1: numeric(12,3)
frequency_2: numeric(12,3)
polarization_1: varchar(20)
polarization_2: varchar(20)
default: boolean
comment: varchar(255)
modified: timestamp
attenuation_1: numeric(10,1)
attenuation_2: numeric(10,1)
channels
PK channel_id: integer (ai)
satellite_id: integer
name: varchar(70)
full_name: varchar(255)
user: varchar(45)
program_title: varchar(70)
comment: varchar(255)
rx_frequency: numeric(12,3)
rx_polarization: varchar(20)
tx_frequency: numeric(12,3)
tx_polarization: varchar(20)
dvb_mode: varchar(10)
nlc_mode: varchar(10)
symbol_rate: numeric(10,4)
fec: varchar(10)
modulation: varchar(20)
rolloff: varchar(10)
pilots: varchar(10)
bit_rate: numeric(10,4)
use_rate: varchar(10)
packet_size: varchar(10)
frame_size: varchar(20)
video_rate: numeric(10,4)
vr_auto: varchar(10)
profile: varchar(20)
eirp: numeric(10,2)
red_eirp: numeric(10,2)
auto_lineup: varchar(10)
audio_rate_1: varchar(10)
audio_rate_2: varchar(10)
audio_rate_3: varchar(10)
audio_rate_4: varchar(10)
audio_channels: integer
decoder_input: varchar(10)
ip_type: varchar(10)
ip_protocol: varchar(10)
ip_source_address: varchar(45)
ip_source_port: integer
ip_destination_address: varchar(45)
ip_destination_port: integer
ip_fec: varchar(10)
ip_buffer_size: integer
encoder_phys_interface: integer
decoder_phys_interface: integer
modified: timestamp
satellite_beacons
PK beacon_id: integer (ai)
satellite_id: integer
frequency: numeric(12,3)
polarization: varchar(20)
default: boolean
comment: varchar(255)
modified: timestamp
satellite_tc
PK tc_id: integer (ai)
satellite_id: integer
frequency: numeric(12,3)
polarization: varchar(20)
attenuation: numeric(10,1)
default: boolean
comment: varchar(255)
modified: timestamp
table_tracking
PK tabletracking_id: integer (ai)
filename: varchar(255)
satellite_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
satellites
PK satellite_id: integer (ai)
name: varchar(45)
operator_id: integer
norad_number: integer
int_designator: varchar(11)
orbit_position: numeric(10,4)
inclination: numeric(10,4)
alias: varchar(45)
modified: timestamp
intelsat_11parameters
PK i11_id: integer (ai)
i11_name: varchar(45)
i11_data: varchar(1000)
satellite_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
mnc_redundancy
PK mnc_id: integer
mnc_name: varchar(40)
enabled: boolean
active_backend
PK backend_id: integer
backend_name: varchar(40)
enabled: boolean
satellite_positions
PK position_id: integer (ai)
satellite_id: integer
azimuth: numeric(10,4)
elevation: numeric(10,4)
polarization: numeric(10,4)
satellite_colocation: varchar(45)
target_number: integer
antenna_id: integer
default: boolean
comment: varchar(255)
modified: timestamp
satellites_operators
PK operator_id: integer (ai)
name: varchar(100)
contact: varchar(2000)
modified: timestamp
norad_tle
PK norad_number: integer
tle_name: varchar(45)
tle_line1: varchar(70)
tle_line2: varchar(45)
default: boolean
comment: varchar(255)
modified: timestamp
1 : 1 (exactly one)
1 : 1 (zero or one)
1 : n (one or more)
1 : n (zero or more)
(ai) auto increment
documents
PK document_id: integer (ai)
file_name: varchar(255)
content_type: varchar(40)
driver: varchar(100)
mnc_name: varchar(40)
comment: varchar(255)
modified: timestamp
inventory_items
PK item_id: integer (ai)
vendor: varchar(100)
serial_no: varchar(40)
model: varchar(40)
driver: varchar(100)
state: varchar(20)
comment: varchar(255)
modified: timestamp
inventory_log
PK entry_id: integer (ai)
item_id: integer
mnc_name: varchar(45)
dev_name: varchar(45)
message: varchar(255)
created: timestamp
inventory_devices
PK device_id: integer (ai)
item_id: integer
mnc_name: varchar(45)
device_name: varchar(45)
state: varchar(20)
online: boolean
modified: timestamp

8.6.1 Table Satellites (satellites)

All satellites known to this system. Number may vary from a dozen to hundreds.

Field	null	Description
satellite_id	no	unique, primary key, auto increment
name	no	free text, typically the name as given by satellite operator
operator_id	yes	1:1 relation to table *satellite-operators*
norad_number	yes	unique name (world wide standardized), 1:1 relation to table *norad_tle*
int_designator	yes	unique name (world wide standardized)
orbit_position	yes	satellite orbit position in degrees
inclination	yes	satellite inclination in degrees
alias	yes	free text, can be used by customer to add own name
modified	no	automatically created timestamp of last change

8.6.2 Table Satellite Operators (satellites_operators)

All satellites operators known to this system. Usually only a small number (2-10) but not limited to this.

Field	null	Description
operator_id	no	unique, primary key, auto generated
name	no	free text, company name of satellite operator
contact	yes	free text, multiline, hotline phone, e-mail etc.
modified	no	automatically created timestamp of last change

8.6.3 Antennas (antennas)

All antennas known to this system. Usually only a small number (2-10) but not limited to this.

Field	null	Description
antenna_id	no	unique, primary key, auto generated

Field	null	Description
devicename	no	free text, max. 11 characters (at the moment) because it is a device name from the MNC system
alias	yes	free text, can be used by customer to add own name
modified	no	automatically created timestamp of last change

8.6.4 Table Satellite Beacons (satellite_beacons)

All satellites beacons known to this system. Normally every satellite has one or more beacon signals, but it is not necessary to store these information because its only needed if you have to track the satellite. So this table could be completely empty.

Field	null	Description
beacon_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this beacon belongs to
frequency	no	beacon frequency, 3 digits precision (MHz)
polarization	no	integer coded receive polarization
attenuation	yes	an attenuation value to be set at the beacon receiver
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

8.6.5 Table Satellite TC (satellite_tc)

All satellite telecommand frequencies / channels known to this system. Every satellite may have one or more TC channels, but these are only required if the application deals with satellite TT&C.

Field	null	Description
tc_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this telecommand channel belongs to
frequency	no	TC signal frequency, 3 digits precision (MHz)
polarization	no	integer coded transmit polarization
attenuation	no	an attenuation value to be set at the TC transmitter

Field	null	Description
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

8.6.6 Table Satellite TM (satellite_tm)

All satellite telemetry frequencies / channels known to this system. Every satellite may have one or more TM channels, but these are only required if the application deals with satellite TT&C.

Field	null	Description
tm_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this telemetry channel belongs to
frequency_1	no	primary receive frequency, 3 digits precision (MHz)
frequency_2	no	secondary receive frequency, 3 digits precision (MHz)
polarization_1	no	integer coded primary receive polarization
polarization_2	no	integer coded secondary receive polarization
attenuation_1	no	attenuation value to be set at the primary telemetry receiver
attenuation_2	no	attenuation value to be set at the secondary telemetry receiver
default	no	boolean, true meaning this is the default beacon for this satellite
modified	no	automatically created timestamp of last change

8.6.7 Table Satellite Positions (satellite_positions)

This tables stores an antenna position (Azimuth/Elevation/Polarization) for a given Satellite and a specific Antenna.

Field	null	Description
position_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite this position belongs to
azimuth	yes	antenna azimuth position in degrees
elevation	yes	antenna elevation position in degrees

Field	null	Description
polarization	yes	antenna polarization position in degrees
satellite-colocation	yes	TODO we need to recap how this field will be used, for now its just free text
target_number	yes	a target number for this satellite to command at the antenna controller (device specific)
antenna_id	no	id of antenna, 1:1 relation to *antenna* table
default	no	boolean, true meaning this is the default satellite position
modified	no	automatically created timestamp of last change

Should later provide a dropdown list with the valid antenna (device) names.

8.6.8 Table Table Tracking (table_tracking)

This tables stores "table tracking" files names for satellites. It could have none or one filename per satellite.

Field	null	Description
tabletracking_id	no	unique, primary key, auto generated
filename	no	free text
satellite_id	no	ID of the satellite this tracking table belongs to
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	automatically created timestamp of last change

8.6.9 Table Intelsat 11 Parameters (intelsat_11parameters)

This tables stores "Intelsat 11 Parameter Sets" for satellites. It could have none or one entry per satellite.

Field	null	Description
i11_id	no	unique, primary key, auto generated
satellite_id	no	ID of the satellite these I11 parameters belong to
i11_name	no	free text, name of I11

Field	null	Description
i11_data	no	free text, (raw) data of I11 (not split into 11 single parameters)
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	timestamp, generated by database at creation time

8.6.10 Table Norad TLE (norad_tle)

This tables stores "Norad Two Line Elements (TLE) " for satellites. It could have none or one entry per satellite.

Field	null	Description
norad_number	no	unique Norad Number, primary key, defined by the satellite
tle_name	no	free text, name of TLE
tle_line1	no	free text, first line of TLE
tle_line2	no	free text, second line of TLE
default	no	boolean, true meaning this is the default table track
comment	yes	free text
modified	no	timestamp, generated by database at creation time

8.6.11 Table Channels (channels)

This tables stores satellite channel records as described below.

Field	null	Description
channel_id	no	A unique ID of this ChannelData record (auto generated).
satellite_id	no	The ID of the satellite this ChannelData record refers to.
name	no	The user defined channel name.

Field	null	Description
full_name	no	The full name of this channel as it appears as a selection in the user interface. The full name is built from the satellite's name, a '-' character and the *name* field in this record. The full name must be unique because the operator selects satellite channels by this name. With POST or PATCH calls to the channel database the backend ignores this field, it automatically generates the *fullName* from the satellite name and *name* .
user	yes	The name of the user who lastly edited the channel
program_title	yes	Name of the program transmitted or received with this channel
comment	yes	A one line comment describing this satellite channel
rx_frequency	yes	The receive frequency in MHz (3 digits precision)
rx_polarization	yes	The receive polarization, one of 'X', 'Y', 'L' or 'R'
tx_frequency	yes	The transmit frequency in MHz (3 digits precision)
tx_polarization	yes	The transmit polarization, one of 'X', 'Y', 'L' or 'R'
dvb_mode	yes	The DVB mode, one of 'DVBS', 'DVBS2', 'NS3' or 'NS4'
nlc_mode	yes	The NLC mode, one of 'OFF' or 'ON'
symbol_rate	yes	The symbol rate in Msym/sec (4 digits precision)
fec	yes	The FEC rate, one of 'AUTO', '1/2', '1/3', '1/4', '2/3', '3/4', '2/5', '3/5', '4/5', '5/6', '6/7', '7/8', '8/9' or '9/10'
modulation	yes	The modulation type, one of 'BPSK', 'QPSK', 'OQPSK', '8PSK', '16PSK', '16QAM', '64QAM', '256QAM', '16APSK' or '32APSK'
rolloff	yes	The roll off factor, one of '0.02', '0.05', '0.10', '0.15', '0.20', '0.25' or '0.35'
pilots	yes	The pilots on/off switch, one of 'OFF' or 'ON'
bit_rate	yes	The data rate in Mbit/sec (4 digits precision)
use_rate	yes	Selects the way the data rate is specified, one of 'SYMBOLRATE' or 'BITRATE'
packet_size	yes	The packet size, one of '188' or '204'

Field	null	Description
frame_size	yes	The frame size, one of 'SHORT' or 'NORMAL'
video_rate	yes	The video data rate in Mbit/sec (4 digits precision)
vr_auto	yes	The video rate auto switch, one of 'FIXED' or 'AUTO'
profile	yes	The video encoding profile, one of 'MPEG2-MP@ML-420', 'MPEG2-MP@LL-420', 'MPEG2-HP@ML-422', 'MPEG2-HP@LL-422', 'MPEG2-SP@LL-420', 'MPEG2-SP@ML-420', 'MPEG2-MP@H14-420', 'MPEG2-MP@HL-420', 'MPEG2-HP@H14-422', 'MPEG2-HP@HL-422', 'MPEG2-HP@HL-420', 'MPEG2-HP@H14-420', 'MPEG2-HP@ML-420', 'MPEG4-BP@L2-420', 'MPEG4-BP@L3-420', 'MPEG4-BP@L4-420', 'MPEG4-MP@L3-420', 'MPEG4-MP@L4-420', 'MPEG4-MP@L4.2-420', 'MPEG4-HP@L3-420', 'MPEG4-HP@L4-420', 'MPEG4-HP@L4.2-420', 'MPEG4-HP@L3-422', 'MPEG4-HP@L4-422' or 'MPEG4-HP@L4.2-422'
eirp	yes	The transmit EIRP in dBW (2 digits precision)
red_eirp	yes	The reduced EIRP for line up in dBW (2 digits precision)
auto_lineup	yes	Configures automatic line up, one of 'OFF' or 'ON'
audio_rate_1	yes	Audio channel 1 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_2	yes	Audio channel 2 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_3	yes	Audio channel 3 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_rate_4	yes	Audio channel 4 data rate, one of 'DISABLED', '64k', '96k', '128k', '160k', '192k', '224k', '256k', '320k' or '384k"
audio_channels	yes	Number of audio channels, (integer, 1-4)
decoder_input	yes	Decoder input, one of 'SAT', 'IP' or 'ASI'
ip_type	yes	IP addressing type, one of 'UNICAST' or 'MULTICAST'
ip_protocol	yes	IP protocol, one of 'UDP', 'RTP', 'TCP' or 'ZIXI'

Field	null	Description
ip_source_address	yes	Source IP address in dotted quad notation
ip_source_port	yes	Source port number (integer, 0-65535)
ip_destination_address	yes	Destination IP address in dotted quad notation
ip_destination_port	yes	Destination port number (integer, 0-65535)
ip_fec	yes	IP FEC selection, one of 'OFF' or 'ON'
ip_buffer_size	yes	Buffer size (integer)
encoder_phys_interface	yes	Encoder physical output number (integer, 1-4)
decoder_phys_interface	yes	Decoder physical input number (integer, 1-4)
modified	no	automatically created timestamp of last change

8.6.12 Table StreamKeys (stream_keys)

This table stores stream keys (BISS keys, SRT passwords) as described below.

Field	null	Description
key_id	no	A unique ID of this StreamKeyData record (auto generated).
name	no	The user defined name of this stream key. This name also must be unique because the operator selects the key by this name.
type	no	The key type, one of 'BISS-1', 'BISS-E' or 'PASSPHRASE'.
key	no	The key value associated with the id and name in this record. For biss keys this is a string entirely consisting of the hexadeximal characters 0123456789ABCDEF, it is either 12 or 16 characters long. For other key types the key is free text.
modified	no	automatically created timestamp of last change

8.6.13 Table BeaconAttenuations (beacon_attenuations)

This table stores the attenuation values for a particular beacon / antenna combination

Field	null	Description
beacon_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.

Field	null	Description
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation	no	The attenuation value (dB)
modified	no	automatically created timestamp of last change

8.6.14 Table TcAttenuations (tc_attenuations)

This table stores the attenuation values for a particular TC channel / antenna combination

Field	null	Description
tc_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation_1	no	The upconverter attenuation value (dB)
attenuation_2	no	The HPA attenuation value (dB)
modified	no	automatically created timestamp of last change

8.6.15 Table TmAttenuations (tm_attenuations)

This table stores the attenuation values for a particular TM channel / antenna combination

Field	null	Description
tm_attn_id	no	A unique ID of this record (auto generated).
beacon_id	no	The ID of the beacon this attenuation belongs to.
antenna_id	no	The ID of the antenna this attenuation belongs to,
attenuation_1	no	The primary TM signal attenuation value (dB)
attenuation_2	no	The secondary TM signal value (dB)
modified	no	automatically created timestamp of last change

8.6.16 Table InventoryItems (inventory_items)

The table inventory_items stores the properties of all items known to the inventory database.

Field	null	Description
item_id	no	A unique ID of this record (auto generated).
serial_no	yes	The serial number of the item / device
vendor	yes	The vendor name (for search filters)
model	yes	The model name (for search filters)
state	no	The administrative state of the device
comment	yes	free text
modified	no	automatically created timestamp of last change

8.6.17 Table InventoryDevices (inventory_devices)

The table inventory_devices stores the device mappings of all items known to the inventory database.

Field	null	Description
device_id	no	A unique ID of this record (auto generated).
item_no	yes	The ID of the item this device is mapped to
mnc_name	yes	The M&C name
device_name	yes	The device name
state	no	The administrative state of the device
online	yes	Boolean flag indicating if the device is online
modified	no	automatically created timestamp of last change

8.6.18 Table InventoryLog (inventory_log)

The table inventory_log stores the live cycle logs of all items which are registered in the inventory_items table.

Field	null	Description
entry_id	no	A unique ID of this record (auto generated).
item_id	no	Links to the inventory_items entry this log entry refers to.

Field	null	Description
mnc_name	yes	The M&C which issued this message or null if the message was entered at the UI
dev_name	yes	The satnms device name which issued this message
message	no	the message text
created	no	timestamp, generated by database at creation time

8.6.19 Table Documents (documents)

The table documents stores information about the documents (pdf, png, jpg) stored in the backend.

Field	null	Description
document_id	no	A unique ID of this record (auto generated).
file_name	no	Name of the file.
content_type	no	Content type of the file. must be one of application/pdf, image/png or image/jpg.
driver	yes	The satnms driver name this document refers to or null if it does not refer to a driver / device type.
mnc_name	yes	The M&C name this document refers to or null if it does not refer to a M&C.
comment	yes	free text
modified	no	automatically created timestamp of last change

The backend handles three types of documents:

- 'M&C related' documents refer to a particular M&C. They have the 'mnc_name' field set.
- 'Driver related' documents refer to a particular device type / driver. They have the 'driver' field set.
- 'global' documents do not refer to anything

Documents are stored on the backend machine in a common directory tree. Global documents are stored at the root of this tree. For each M&C and for each device driver the backend creates a subdirectory. M&C or driver related documents are stored in one of these subdirectories. File names must be unique within the subdirectory where they are are stored.

8.6.20 Table GlobalSettings (globalSettings)

The global_settings stores the properties of changed settings from default by admins.

Field	null	Description
settings_id	Integer	A Unique Id of an setting
key	no	Setting name as key
value	no	Value of the setting
username	yes	Username to which the setting belongs
isLocked	no	Whether the setting is locked for user to overwrite it
modified	no	timestamp, generated by database at creation time

8.6.21 Table User Settings (userSettings)

The user_settings stores the properties of changed settings from default by user.

Field	null	Description
settings_id	Integer	A Unique Id of an setting
key	no	Setting name as key
value	no	Value of the setting
username	yes	Username to which the setting belongs
isLocked	no	Whether the setting is locked for user to overwrite it
modified	no	timestamp, generated by database at creation time

8.7 Fonts

The satnms software uses a number of predefined fonts to be used in the user interface. The table below shows which font types / sizes match the pixel by pixel position and size values of existing screen definitions.

Name	Font	Size	Style
small	Arial	10px	PLAIN
plain	Arial	12px	PLAIN
bold	Arial	12px	BOLD

Name	Font	Size	Style
title	Arial	16px	BOLD
huge	Arial	28px	PLAIN
typewriter	Courier New	10px	PLAIN

9 Device Driver Descriptions

The following pages describe all device drivers coming with the sat-nms software. Each device driver description contains a short paragraph about what type of equipment this driver is for and how should be configured that it works seamlessly with the device.

This manual contains three tables which give you fast access to the device driver documentation:

- Device Drivers By Name --- contains a table which lists all device drivers sorted by their driver name.
- Device Drivers By Vendor --- contains a table which lists all device drivers sorted by the name of the vendor of the device to control
- Device Drivers By Tag --- contains a table which lists all device drivers sorted by tags assigned to the device drivers. These tags classify devices to groups like HPAs, antenna controllers, IRDs and more.

9.1 Device Drivers By Name

Driver	***Version***	***Vendor***	***Tags***
Acorde-ACRS-IDU	1.00 250916	Acorde Technologies	Redundancy-Controller LNA/LNB-Controller
Acorde-BUC-ACCVT	1.06 240807	Acorde Technologies	Block-Upconverter
ACTi-ACM1231-Webcam	1.03 171212	ACTi	Camera
Actia-Sodielec-DNT-HPA	1.08 240314	Actia-Sodielec	HPA TWTA
Actia-Sodielec-DSS1300	1.02 171212	Actia-Sodielec	I/O-Controller
Adtec-Afiniti-Decoder	1.01 220423	Adtec	IRD
Adtec-Afinity-Encoder	1.03 230202	Adtec	Encoder
Adtec-EN81	1.11 210916	Adtec	Encoder Modulator
Adtec-RD60	1.08 171212	Adtec	IRD

Driver	Version	Vendor	Tags
Advantech-ADAM5K	1.02 171212	Advantech	I/O-Controller
Advantech-ADAM6K	1.01 171212	Advantech	I/O-Controller
Advantech-Genesis	1.01 241212	Advantech Wireless	HPA SSPA
Advantech-SSPB-GCP	1.01 240314	Advantech Wireless	HPA SSPA
Advantech-SSPB-MCP	1.01 240314	Advantech Wireless	HPA SSPA
Advantech-SSPB-SNMP	1.06 250715	Advantech Wireless	HPA SSPA
Advantech-SSPB	1.09 190725	Advantech Wireless	HPA SSPA
Agilent-Spectrum-Analyzer	1.04 171220	Agilent	Spectrum-Analyzer Measurement
Agilis-BUC	1.01 190725	Agilis	Block-Upconverter
Altec-Puregas-P210	1.00 240514	Altec Air	Dehydrator
Altec-Puregas-P550	1.01 240314	Altec	Dehydrator
Anacom-KuBand-Transceiver	1.05 240314	Anacom	Transceiver
Anacom-Protection-Switch	1.01 171212	Anacom	Redundancy-Controller
Andrew-APC100	1.04 171220	Andrew	Antenna-Controller
Antech-Deice	1.01 171220	Antech	Deice-Controller
Antenna-BBU-Connector	1.00 230228	SatService	logical

Driver	Version	Vendor	Tags
Antenna-Dummy	01.00 200307		
Antenna-Fixed	1.00 250401	SatService	logical
Antenna-Management	1.09 251126	SatService	logical
Antenna-Pointing-Fixed	1.03 171221	SatService	logical
Antenna-Pointing-Hiltron	1.00 210124	SatService	logical
Antenna-Pointing-ODM	2.00 181004	SatService	logical
Antenna-Pointing	3.04 171221	SatService	logical
Antenna-Proxy	1.01 171221	SatService	logical
Antenna-Tracking	1.06 171221	SatService	logical
Antenna-Wind-Protection	1.01 210427	SatService	logical
AntwerpSpace-G3-Demod	1.09 230310	Antwerpspace	Demodulator
AntwerpSpace-G3-Mod	1.08 230310	Antwerpspace	Modulator
AntwerpSpace-IBB-Modem	1.15 240314	Antwerpspace	Modem
APC-ATS	1.02 240314	APC	Power-Module
APC-RPDU	1.03 190130	APC	Power-Module
APKieffer-TWTA-AC	1.02 240314	AP-Kieffer	Air-Conditioning

Driver	Version	Vendor	Tags
Apollo-Switch	1.02 171212	Apollo	Redundancy-Controller
APS-Backup-Antenna-Control	2.01 171221	SatService	logical
APS-Backup-Antenna-Switch	2.01 171221	SatService	logical
APS-Site-Switch	1.05 171221	SatService	logical
ARD-Backup-Client	1.04 250318	SatService	logical
ARD-Backup-Server	1.03 250318	SatService	logical
Arithmetic	1.01 171221	SatService	logical
ASCSignal-NGC-ACU	1.07 190725	ASCSignal	Antenna-Controller
ASCSignal-NGC-BDC	1.02 200309	ASCSignal	Block-Downconverter
ASCSignal-NGC-Deice	1.00 190725	ASCSignal	Deice-Controller
ASCSignal-NGC-Hub-AC	1.00 190725	ASCSignal	Air-Conditioning
ASCSignal-NGC-RedundancyModule	2.01 240314	ASCSignal	Redundancy-Controller LNA/LNB-Controller
ASCSignal-NGC-Temperature	1.00 190725	ASCSignal	Temperature-Sensor
ASCSignal-NGC-Weather	1.00 190725	ASCSignal	Weather
Astro-EdgeStreamer	1.05 250708	ASTRO Strobel Kommunikationssysteme GmbH	IRD
Ateme-DR5000	3.01 250606	Ateme	IRD

Driver	Version	Vendor	Tags
Ateme-DR8400	1.10 240314	Ateme	IRD
Ateme-Kyrion-CM5000	1.25 250526	Ateme	Encoder Modulator
Ateme-TitanEdge-Bridge	1.02 251203	Ateme	Video-Converter
Ateme-TitanEdge-Encoder	1.11 251203	Ateme	Encoder
Ateme-TitanEdge-Gateway	1.11 251203	Ateme	Gateway
Ateme-TitanEdge-IRD	2.22 251126	Ateme	IRD
Aten-VGA-Splitter	1.01 171212	Aten	Graphics
AtlanTecRF-TLT	1.02 220929	Atlantic-Microwave	Testloop-Translator
AtlanticSat-ASC300	1.02 230911	Atlantic Satellite Corporation	Tracking-Receiver
Auto-RollOff	1.02 171221	SatService	logical
AutomationDirect-Terminator	2.00 190312	Automation-Direct	I/O-Controller
AvL-ITC-ACU	1.07 251104	AvL	Antenna-Controller
AvL-TracStar	1.08 180323	AvL Technologies	Antenna-Controller
AzElCollector	1.02 171221	SatService	logical
Band-Pol-4-To-1	1.01 171221	SatService	logical
Barco-RE4220	1.03 240314	Barco	Encoder

Driver	Version	Vendor	Tags
Bard-MC4002-AC	1.00 231130	Bard	Air-Conditioning
BISS-KeySelect-CDB-DBonly	2.04 171221	SatService	logical
BISS-KeySelect-CDB	2.04 171221	SatService	logical
BISS-KeySelect	1.00 240213	SatService	logical
Blankom-DRP393-SNMP	1.07 240314	Blankom	IRD
BMS-IOM-AHU	1.03 240314	BMS-IOM	Measurement
BMS-IOM-Generator	1.01 1800810	BMS-IOM	Generator
BMS-IOM-Powermeter	1.01 180810	BMS-IOM	Powersensor
BMS-IOM-Socomec-Powermeter	1.01 180810	BMS-IOM	Powersensor
BMS-IOM-Trend-DI	1.01 180810	BMS-IOM	I/O-Controller
BMS-UPS	1.01 180813	BMS-IOM	UPS
Brainbox-EDxxx	1.01 240314	Brainbox	I/O-Controller
Calian-Decimator	1.06 240918	Calian	Spectrum-Analyzer Measurement
Calian-MCI-ACU	1.05 250909	Calian	Antenna-Controller
CandM-Switch-30510	1.00 220928	CandM	Redundancy-Controller
CandM-Switch-308901	1.00 220621	CandM	Redundancy-Controller

Driver	Version	Vendor	Tags
CandM-Switch	1.02 210819	CandM	Redundancy-Controller
Carrier-Site-Switch	1.07 221102	SatService	logical
CGC-ACU	1.18 240611	Comtech CGC	Antenna-Controller
Channel-Select	1.01 171221	SatService	logical
ChinaSuncom-9293S3-ACU	1.00 231025	China Suncom	Antenna-Controller
ChinaSuncom-Beacon-Receiver-SC9800	1.00 231030	China Suncom	Tracking-Receiver
Choice-Translator	2.00 230116	SatService	logical
Choice	1.05 171221	SatService	logical
Cibred-LAB4	1.01 190515	Cibred Sud	Dehydrator
Cisco-2800GSM-SNMP	1.03 171212	Cisco	Router
Cisco-2900-SNMP	1.04 171212	Cisco	Router
Cisco-D9800-IRD	1.04 210608	Cisco Synamedia	IRD
Cisco-Monitor	1.02 171212	Cisco	Router
Cisco-Router	1.01 171212	Cisco	Router
Codan-7586L	1.010 230323	Codan CPI	Redundancy-Controller
Codan-BUC-7586L	1.01 230323	Codan	Block-Upconverter

Driver	Version	Vendor	Tags
Codan-LBUC-6x00	2.04 240314	Codan	Block-Upconverter
Codan-Transceiver	1.04 240314	Codan	Block-Upconverter Block-Downconverter
Collector-Two-Way	1.05 240528	SatService	logical
Collector	1.06 171221	SatService	logical
ColumbiaWeather-MicroServer	1.00 250714	Vaisala	Weather
Comment	1.01 171221	SatService	logical
Compare	1.03 171221	SatService	logical
Comstream-ABR200	1.01 171212	Comstream	IRD
Comstream-ABR202	1.05 171212	Comstream	IRD
Comstream-CM601A-SCPC	1.06 190725	Comstream	Modem
Comstream-CM701-MC	1.02 190725	Comstream	Modem
Comstream-CM701	1.02 190725	Comstream	Modem
Comstream-CX101	0.01 171212	Comstream	Redundancy-Controller
Comstream-CX801	1.02 171212	Comstream	Redundancy-Controller
Comstream-DAC700-PRE	1.04 171212	Comstream	Encoder Decoder
Comstream-DAC700	1.03 171212	Comstream	Encoder Decoder

Driver	Version	Vendor	Tags
Comstream-Demodulator	1.01 171212	Comstream	Demodulator
Comstream-LoSpeed-Modem-SCPC	1.05 210916	Comstream	Modem
Comstream-Modem-Card-SCPC	1.03 210916	Comstream	Modem
Comstream-Modulator	1.03 210917	Comstream	Modulator
Comtech-CDD564	1.00	Comtech	Modem
Comtech-CDM570	1.06 200420	Comtech	Modem
Comtech-CDM600	1.53 240314	Comtech	Modem
Comtech-CDM760	1.00 241210	Comtech	Modulator Demodulator
Comtech-DT4500-SNMP	1.02 200116	Comtech	Downconverter
Comtech-DT4500	1.06 200116	Comtech	Downconverter
Comtech-LBC4000	1.00 250428	Comtech	Upconverter Downconverter
Comtech-LPOD-SNMP	1.05 240314	Comtech	Block-Upconverter
Comtech-MBT5000	1.02 240314	Comtech	Block-Upconverter Block-Downconverter
Comtech-MBT5000A	1.00 250515	Comtech	Block-Upconverter Block-Downconverter
Comtech-UT4500-SNMP	1.02 190725	Comtech	Upconverter
Comtech-UT4500	1.07 250724	Comtech	Upconverter

Driver	*Version*	*Vendor*	*Tags*
ComWinTop-THxxS	1.02 250910	ComWinTop	Temperature-Sensor
Conditional-Replicator	1.00 220201	SatService	logical
CPI-350W-TWTA	2.05 240314	CPI	HPA TWTA
CPI-CHPA	1.09 241009	CPI	TWTA HPA
CPI-GaN-SSPA	1.01 240314	CPI	SSPA HPA
CPI-KHPA-G3	2.11 240313	CPI	HPA Klystron
CPI-KHPA-G4	2.15 250702	CPI	HPA Klystron
CPI-LPA	2.06 240313	CPI	HPA TWTA
CPI-Mini-ODU	1.27 241119	CPI	TWTA HPA
CPI-SLPA	1.11 250610	CPI	HPA TWTA
CPI-SRCU-BCIP	1.01 240314	CPI	Redundancy-Controller
CPI-SRCU	1.00 190326	CPI	Redundancy-Controller
CPI-Switch	1.03 210819	CPI	Redundancy-Controller
CPI-TouchPower	1.13 240917	CPI	TWTA HPA
CPI-TWTA-Ka	2.07 240314	CPI	HPA TWTA
CPI-TWTA	2.06 240314	CPI	HPA TWTA

Driver	Version	Vendor	Tags
CPI-VZU-TWTA	1.04 240314	CPI	HPA TWTA
Cross-Frequency-Source	1.01 171213	Cross Technology	Frequency-Reference
Cross-UpDownconverter	1.03 190725	Cross Technology	Upconverter Downconverter
CSM-Spectrum-Analyzer	3.04 240104	Various	Spectrum-Analyzer Measurement
Dantherm-ComboCooling1000	1.01 250610	Dantherm	Air-Conditioning
Dantherm-TKS3000A	1.02 171213	Dantherm	Air-Conditioning
De-Icing-Buttons	1.01 180608	SatService	logical
De-Icing-Multiplex	2.01 180720	SatService	logical
Dektec-DTA2137	1.07 130408	Dektec	Demodulator
Dektec-DTA2137	3.07 200114	Dektec	Demodulator
Delay-Line-Mirror	1.01 171221	SatService	logical
Delay-Line	1.01 171221	SatService	logical
Delayed-Gate-3	1.02 250207	SatService	logical
DEV-4111-DA	1.01 231025	DEV Systemtechnik	Distribution-Amplifier
DEV-Distribution-2190	1.02 171006	DEV Systemtechnik	Distribution-Amplifier
DEV-FO-RX	1.00 201019	DEV Systemtechnik	Optical-Links

Driver	Version	Vendor	Tags
DEV-FO-TX	1.00 201019	DEV Systemtechnik	Optical-Links
DEV-Frame	1.00 201019	DEV Systemtechnik	Main-Frame
DEV-Matrix-1986	1.06 240314	DEV Systemtechnik	Switch-Matrix
DEV-Matrix-1996	1.00 250827	DEV Systemtechnik	Switch-Matrix
DEV-Switch-2-To-1	1.00 201202	DEV Systemtechnik	Switch
Device-Copy	1.01 171221	SatService	logical
Device-Mirror	1.01 171221	SatService	logical
Device-Switcher	1.00 20220629	SatService	logical
DoubleD-DDA89	1.00 251114	DoubleD Electronics	Redundancy-Controller
Downconverter-Protection-Controller	1.00 250318	SatService	logical
Dual-Timezone-Clock	1.00 190528	SatService	logical
Dual-Translator	1.01 171221	SatService	logical
Dual-Wind-Speed	1.00 210427	SatService	logical
Dummy-Matrix	0.00 000000		
E2V-TWTA	1.06 200728	E2V	HPA TWTA
ECU-J0475	1.00 230810	ECU Corporation	Air-Conditioning

Driver	Version	Vendor	Tags
EF-Data-SDM300	1.02 190725	EF-Data	Modem
Emerson-AC-HPM	1.01 171213	Emerson Liebert	Air-Conditioning
Endrun-FDC3300	1.02 170926	Endrun Technologies	Distributor
Enensys-Vyper-Modulator	1.05 251104	Enensys	Modulator
Envivio-4Caster-B3	1.04 171213	Envivo	Encoder
Envivio-4View-D1	1.02 171213	Envivo	IRD
Ericsson-AVP	1.06 250224	Ericsson	Encoder Modulator
Ericsson-Voyager-II-v9.5	3.19 210916	Ericsson	Encoder Modulator
Ericsson-Voyager-II	4.03 210916	Ericsson	Encoder Modulator
ET-ADH-2A	1.02 210519	Environmental Technology	Dehydrator
ETI-ADH-NETCOM	1.07 231220	Environmental Technology	Dehydrator
ETI-ASM-1-Manifold	1.01 231103	Environmental Technology	Dehydrator
ETL-2773-XXYY	1.01 240314	ETL Systems	LNA/LNB-Controller
ETL-Alto-Amp	1.04 180814	ETL Systems	LNA/LNB-Controller Amplifier
ETL-Alto-DA	1.02 240314	ETL Systems	Distribution-Amplifier
ETL-Dextra-Combiner	1.01 240314	ETL Systems	Combiner

Driver	Version	Vendor	Tags
ETL-Dextra-DistAmp	1.07 190717	ETL Systems	Distribution-Amplifier
ETL-Genus-100MHz	1.00 251020	ETL Systems	Frequency-Reference
ETL-Genus-10MHz	1.00 240315	ETL Systems	Frequency-Reference
ETL-Genus-ALT	1.00 230526	ETL Systems	Splitter
ETL-Genus-COM402	1.02 221108	ETL Systems	Combiner
ETL-Genus-DA-22512	1.07 240103	ETL Systems	Distribution-Amplifier
ETL-Genus-DIV401	1.02 210521	ETL Systems	Splitter
ETL-Genus-Downconverter	1.01 251020	ETL Systems	Downconverter
ETL-Genus-Frame	1.05 250217	ETL Systems	Main-Frame
ETL-Genus-PRN	1.01 250925	ETL Systems	LNA/LNB-Controller
ETL-Genus-SRY-RX	1.00 240826	ETL Systems	Optical-Links
ETL-Genus-SRY-TX	1.01 250606	ETL Systems	Optical-Links
ETL-Genus-Upconverter	1.01 251020	ETL Systems	Upconverter
ETL-Genus-VTR-101-Matrix	1.02 240428	ETL Systems	Switch-Matrix
ETL-Genus-VTRC-101-Matrix	1.02 240428	ETL Systems	Switch-Matrix
ETL-HMI-Reboot	1.00 230614		

Driver	*Version*	*Vendor*	*Tags*
ETL-Hurricane-Matrix	1.01 190403	ETL Systems	Switch-Matrix
ETL-LS-Switch	1.09 241213	ETL Systems	Switch-Controller
ETL-MatrixPort-Emulator	1.01 250207	SatService	logical
ETL-Modular-Frame-26128	1.04 180429	ETL Systems	Main-Frame
ETL-NGM100-Matrix	2.07 250527	ETL Systems	Switch-Matrix
ETL-NGM100C-Matrix	1.02 250519	ETL Systems	Switch-Matrix
ETL-NGM21C-Matrix	1.12 250519	ETL Systems	Switch-Matrix
ETL-NGM30-Matrix	1.13 250519	ETL Systems	Switch-Matrix
ETL-NSN100-Matrix	4.02 250527	ETL Systems	Switch-Matrix
ETL-PDSwitch-Emulator	4.00 231211	SatService	logical
ETL-Piranha	1.01 240314	ETL Systems	LNA/LNB-Controller
ETL-POE-Switch	1.01 240314	ETL Systems	Switch-Controller
ETL-SHF-Switch-2to2	1.00 250428	ETL Systems	Switch-Controller
ETL-SHF-Switch	1.02 171220	ETL Systems	Switch-Controller
ETL-Splitter-DIV409	1.03 171220	ETL Systems	Splitter
ETL-Splitter-DIV429	1.00 190920	ETL Systems	Splitter

Driver	Version	Vendor	Tags
ETL-Stingray-FO-Base-Serial	1.00 201020	ETL Systems	Distribution-Amplifier
ETL-Stingray-FO-Base	1.00 200922	ETL Systems	Distribution-Amplifier
ETL-Stingray-FO-RX-Serial	1.00 201020	ETL Systems	Optical-Links
ETL-Stingray-FO-RX	1.01 211210	ETL Systems	Optical-Links
ETL-Stingray-FO-TX-Serial	1.00 201020	ETL Systems	Optical-Links
ETL-Stingray-FO-TX	1.01 211210	ETL Systems	Optical-Links
ETL-SwitchUnit-23258	1.03 180814	ETL Systems	Switch-Controller
ETL-Volcano-Matrix	1.03 240314	ETL Systems	Switch-Matrix
ETL-VTR70-Matrix	1.04 251006	ETL Systems	Switch-Matrix
ETL-VTR70C-Matrix	1.03 251021	ETL Systems	Switch-Matrix
ETL-VTX100-Matrix	1.03 240702	ETL Systems	Switch-Matrix
Evertz-7700-Frame	1.01 250901	Evertz	Main-Frame
Evertz-7703-BPX	1.02 190716	Evertz	Optical-Links
Evertz-7707-ET	1.00 181206	Evertz	Optical-Links
Evertz-7707-LR	1.01 190321	Evertz	Optical-Links
Evertz-7707-LT	1.01 190321	Evertz	Optical-Links

Driver	Version	Vendor	Tags
Evertz-7708-LR	1.01 190321	Evertz	Optical-Links
Evertz-7708-LT	1.00 190321	Evertz	Optical-Links
Evertz-7800-Frame	1.02 240314	Evertz	Main-Frame
Evertz-7881-IRD	1.02 240314	Evertz	IRD
Evertz-EMR-Matrix	1.00 250204	Evertz	Switch-Matrix
Execute	2.0 211025	SatService	logical
Fanuc-PLC9030	1.03 171213	Fanuc	I/O-Controller
Fault-Array	1.01 171221	SatService	logical
Fault-Gate-3	1.05 171221	SatService	logical
Fault-Gate	1.04 171221	SatService	logical
FeiZyfer-GPS-Receiver	1.03 250703	FeiZyfer	GPS Frequency-Reference
File-Recorder	2.03 190616	SatService	logical
Filter	1.02 171221	SatService	logical
Floating-Limit	1.01 200519	SatService	logical
Foxcom-SatLight-Platinum	1.12 190515	Foxcom	Optical-Links
Frequency-Loop	1.04 171221	SatService	logical

Driver	Version	Vendor	Tags
Frequency-Split	1.01 171221	SatService	logical
Gate-Array-20	1.00 190528	SatService	logical
Gate-Array	1.03 190506	SatService	logical
GDSatcom-ACU930	1.01 250327	GD Satcom	Antenna-Controller
GDSatcom-ACU940	1.05 250521	GD Satcom	Antenna-Controller
GDSatcom-ACU950-MP	1.03 250407	GD Satcom	Antenna-Controller
GDSatcom-ACU950	1.14 250407	GD Satcom	Antenna-Controller
GDSatcom-ACU970	1.06 250407	GD Satcom	Antenna-Controller
GDSatcom-ACU980	1.01 250407	GD Satcom	Antenna-Controller
GDSatcom-DTR520	1.03 240916	GD Satcom	Tracking-Receiver
GDSatcom-Rack-SSPA	1.00 240314	GDSatcom CPI	SSPA
GDSatcom-RSC1200	2.00 220317	GD Satcom	Redundancy-Controller LNA/LNB-Controller
GDSatcom-SSPA-QuadMod	1.02 240314	GDSatcom	SSPA
Generex-CS121-UPS-Adapter	1.01 171213	Generex	UPS
Generic	1.01 171221	SatService	logical
Geosync-PS	1.00 181105	Geosync Microwave	Power-Supply

Driver	Version	Vendor	Tags
Geosync-RCU-R2R	1.01 210924	Geosync Microwave	Redundancy-Controller
Geosync-RCU	1.07 251030	Geosync Microwave	LNA/LNB-Controller Redundancy-Controller
Geosync-TLT	1.04 190725	Geosync Microwave	Testloop-Translator
Gude-ExpertPowerControl	1.01 171213	Gude	Power-Module
GundD-CompactCenter	1.01 171213	Guntermann und Drunck	Graphics
Haivision-Makito-X-Decoder	1.04 250910	Haivision	Decoder
Haivision-Makito-X	1.07 251112	Haivision	Encoder
Haivision-SRT-Gateway	1.07 251104	Haivision	Gateway
Harris-Panacea	1.02 171213	Harris	Switch-Matrix
Hilkom-BNova-SNMP	1.02 171213	Hilkom	IRD
Hiltron-ACU2	1.07 250514	Hiltron	Antenna-Controller
Hiltron-Deice	1.03 180115	Hiltron	Deice-Controller
Hiltron-HCS-Frame	1.00 180110	Hiltron	Main-Frame
Hiltron-HCS-Power	1.01 171213	Hiltron	Optical-Links LNA/LNB-Controller
Hiltron-HCS-PS	1.01 180117	Hiltron	Power-Supply
Hiltron-HCS-SSPA	1.03 190725	Hiltron	HPA SSPA

Driver	Version	Vendor	Tags
Hiltron-HCS-Switch	1.01 171213	Hiltron	Redundancy-Controller
Hiltron-HCS-TWTA	1.06 200728	Hiltron	HPA TWTA
Hiltron-Multi-Deice	1.00 201023	Hiltron	Deice-Controller
Hitec-AFP-Deice	1.02 171213	Hitec	Antenna-Controller
Hitec-AFP-Deice	1.02 171213	Hitec	Deice-Controller
Hitec-HACU-01-NAOS	1.22 230427	Hitec	Antenna-Controller
Hitec-HACU-01-V3.02	1.00 250723	Hitec	Antenna-Controller
Hitec-HACU-01-V3	2.01 250923	Hitec	Antenna-Controller
Hitec-HACU-01	1.09 211027	Hitec	Antenna-Controller
Hitec-HAFP-00-NAOS1	1.04 230418	Hitec	Deice-Controller
Hitec-HAFP-00	1.04 190228	Hitec	Deice-Controller
Hitec-HSCU-00	1.05 251126	Hitec	Deice-Controller
Hitec-NTP	1.0 201124	SatService	logical
HP-Powermeter-Legacy	1.00 201021	Hewlett-Packard	Powersensor Measurement
HP-Powermeter-SCPI	1.02 171220	Hewlett-Packard	Powersensor Measurement
HP-Synthesizer-SCPI	1.03 220823	Hewlett-Packard	Synthesizer

Driver	Version	Vendor	Tags
HWGroup-Poseidon2	1.01 230207	HW Group	Temperature-Sensor I/O-Controller
HWGroup-STE2	1.03 180223	HW Group	Temperature-Sensor
HWGroup-STE	1.01 230209	HW Group	Temperature-Sensor
IDirect-5000	1.06 190725	iDirect	Modem
Infratec-PM	1.03 171213	Infratec	Power-Module
Inmarsat-NLES-Status	1.01 250917	SatService	logical
Input	1.02 171221	SatService	logical
IO-FEP2-Monitor	1.00 231213	SatService	logical
IP-Sens-IPS2225P	1.00 240411	IP Sens	Temperature-Sensor
IRD-Chain	1.02 171221	SatService	logical
Irt-Pico-BUC-RCP	1.01 220816	IRT Technologies Spacepath	Block-Upconverter
Irt-Pico-BUC-RCP	1.06 190725	IRT Technologies	Block-Upconverter
Janitza-UMG604	1.02 171220	Janitza	Powersensor Measurement
KBR-Multimess-4F96	1.02 190212	KBR	Powersensor Measurement
Key-Generator	1.01 230322		
Keysight-FieldFox	1.1 250319	Keysight	Measurement Spectrum-Analyzer

Driver	Version	Vendor	Tags
Leitch-Panacea	1.01 171213	Leitch	Switch-Matrix
Level-Add	1.07 240822	SatService	logical
Level-AddRed	1.07 240822	SatService	logical
Level-Control	2.04 200310	SatService	logical
Level-Monitoring	1.00 190523	SatService	logical
Level-Set	2.08 200310	SatService	logical
LineUp-Control	1.01 171221	SatService	logical
Link-Mirror	1.02 171221	SatService	logical
Link-Static	1.02 171221	SatService	logical
Link-Two-Way	1.01 171221	SatService	logical
Link	2.02 171221	SatService	logical
Log-Test	1.00 201015		
Logger	1.01 171221	SatService	logical
Logical-Gate-3	3.00 251029	SatService	logical
Logical-Gate	3.00 251029	SatService	logical
LPT-3000RX4	1.0 250326	LPT	Measurement Spectrum-Analyzer

Driver	Version	Vendor	Tags
Lynx-DVD5420-ASI-Distr	1.02 171213	Lynx Technik	Distributor
Lynx-DVD5820-ASI-Distr	1.02 171213	Lynx Technik	Distributor
Lynx-DVD5820-Main	1.03 171213	Lynx Technik	Main-Frame
Macro-Player	1.03 171221	SatService	logical
Matrix-Protection-Switcher	1.01 240313	SatService	logical
Matrix-Site-Switch	2.01 190930	SatService	logical
Maxtech-LRC-1200	1.04 220422	Maxtech	LNA/LNB-Controller Redundancy-Controller
MCL-MT-TWTA-CSP	1.10 250107	MCL	HPA TWTA
MCL-MT-TWTA	2.05 200728	MCL	HPA TWTA
Measurement-Converter	1.01 171221	SatService	logical
Mediakind-CE1	1.01 250326	Mediakind	Encoder
Mediakind-RX1	1.02 250526	Mediakind	IRD
Meinberg-LantimeM300	1.02 221110	Meinberg	GPS Frequency-Reference
Meinberg-SDU	1.00 240411	Meinberg	Distribution-Amplifier
Mesa-WNT	1.02 180223	Mesa	Weather
Mesa-WSWD	1.02 180223	Mesa	Weather

Driver	Version	Vendor	Tags
MFM-IpDecoder	1.00 140903		
MFM-IpEncoder	1.00 140902		
MFM-Ird	1.04 140924		
MFM-PlayList	1.02 141029		
MFM-SkyWANIdu	1.00 140828		
MFM-Upc	1.03 140827		
MIB2-Ethernet-Switch	1.01 240523	Generic	Interface-Monitor
MIB2-Interface-Monitor	1.04 191010	Generic	Interface-Monitor
MIB2-Storage-Monitor	1.01 250224	Generic	Storage-Monitor
MIB2-UPS-Monitor	1.04 250428	Generic	UPS
MiniCircuits-Switch-ZT	1.02 240314	MiniCircuits	Switch-Controller
Mirad-VPC	1.03 171213	Mirad Microwave	Switch-Controller Variable-Power-Combiner
MissionMicrowave-SSPA	0.06 240729	Mission Microwave	Block-Upconverter SSPA
Mitec-ARK-RX	1.00 190710	Mitec	LNA/LNB-Controller Redundancy-Controller
Mitec-Switch	1.02 171213	Mitec	Switch-Controller Variable-Power-Combiner

Driver	Version	Vendor	Tags
Mitec-TX	1.01 240314	Mitec	HPA SSPA
Miteq-Downconverter-9900-SNMP	1.00 230126	Miteq	Downconverter
Miteq-Downconverter-9900	1.00 211008	Miteq	Downconverter
Miteq-DVM100	1.03 210916	Miteq	Modulator
Miteq-KBand-Upconverter	1.02 190725	Miteq	Upconverter
Miteq-KuBand-Upconverter	1.00 210512	Miteq	Upconverter
Miteq-LBand-Downconverter	1.03 200116	Miteq	Downconverter
Miteq-LBand-Upconverter	1.03 190725	Miteq	Upconverter
Miteq-NSU	1.04 250718	Miteq	Upconverter Redundancy-Controller
Miteq-TLT-LO	1.00 210517	Miteq	Testloop-Translator
Miteq-TLT	1.01 210928	Miteq	Testloop-Translator
Miteq-UPC	1.03 251230	Miteq	Uplink-Power-Control
Miteq-Upconverter-9900-SNMP	1.00 230126	Miteq	Downconverter
Modcod-Splitter	1.01 171221	SatService	logical
Modem-Protection-Controller	1.05 250319	SatService	logical

Driver	Version	Vendor	Tags
Monitor-Switch	1.02 171221	SatService	logical
Monterey-140	1.01 171214	Chaparral Communications	Antenna-Controller
Moxa-ioLogik-1200	1.03 240314	Moxa	I/O-Controller
Moxa-ioLogik-2214	1.00 240402	Moxa	I/O-Controller
Moxa-ioLogik-4000	1.02 171124	Moxa	I/O-Controller
Moxa-ioLogik-4200	1.03 171124	Moxa	I/O-Controller
Moxa-ioLogik-Current	1.01 171124	Moxa	I/O-Controller
Moxa-ioLogik-Temperature	1.01 171124	Moxa	I/O-Controller Temperature-Sensor
Moxa-ioThinx-4510	1.00 210608	Moxa	I/O-Controller
Multi-Add	1.00 250826	SatService	logical
Multi-Protection-Switch	1.02 171221	SatService	logical
Multi-Site-Switch	1.01 210211	SatService	logical
Multi-Symmetric-Site-Switch	1.00 250805	SatService	logical
N-Way-Switch-Latching	2.03 171221	SatService	logical
N-Way-Switch	1.03 171221	SatService	logical
NAIS-FP-8-Port-Switch	1.01 170926	NAIS/Panasonic	I/O-Controller

Driver	*Version*	*Vendor*	*Tags*
NAIS-FP-AD-Test	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-AD4-2Channel	1.0 210316	NAIS/Panasonic	I/O-Controller
NAIS-FP-Alarm-Circuit	1.03 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Inv-Alarm-Circuit	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Inv-Pulse-Output	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Latching-Relay	1.03 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Protection-Switch	1.04 171214	NAIS/Panasonic	Redundancy-Controller
NAIS-FP-Pulse-Output	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Relay	1.02 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-RTD-Module	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Simple-Input	1.02 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Simple-Output	1.03 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-TxOn-Switch	1.01 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP-Watchdog-Output	1.02 171214	NAIS/Panasonic	I/O-Controller
NAIS-FP0-REG	1.03 200113	NAIS/Panasonic	I/O-Controller
NAIS-FP0-RO	1.02 200113	NAIS/Panasonic	I/O-Controller

Driver	Version	Vendor	Tags
NAIS-FP0	1.02 171214	NAIS/Panasonic	I/O-Controller
Narda-MCP	1.03 171220	Narda	Spectrum-Analyzer Measurement
Narda-NRA	1.01 250910	Narda	Measurement Spectrum-Analyzer
Narda-Upconverter	1.03 190725	Narda	Upconverter
NDSatCom-1Axis-ACU	1.01 171214	NDSatCom	Antenna-Controller
NDSatCom-3Axis-ACU	1.02 171214	NDSatCom	Antenna-Controller
NDSatCom-ACU4100A-SNG	1.01 171214	NDSatCom	Antenna-Controller
NDSatCom-ACU4100B	1.02 171214	NDSatCom	Antenna-Controller
NDSatCom-ACU5000-SNG	3.11 240314	NDSatCom	Antenna-Controller
NDSatCom-ACU5030	1.05 180226	NDSatCom	Antenna-Controller
NDSatCom-ACU6030-REST	1.02 250212	NDSatCom	Antenna-Controller
NDSatCom-CPI	1.07 240314	NDSatCom	HPA TWTA
NDSatCom-KBand-Downconverter	2.07 200116	NDSatCom	Downconverter
NDSatCom-KBand-Upconverter	2.06 230715	NDSatCom	Upconverter
NDSatCom-KuBand-Downconverter	2.07 200116	NDSatCom	Downconverter
NDSatCom-KuBand-Upconverter	2.06 230715	NDSatCom	Upconverter

Driver	Version	Vendor	Tags
NDSatCom-KuHBand-Downconverter	2.07 200116	NDSatCom	Downconverter
NDSatCom-KuHBand-Upconverter	2.06 230715	NDSatCom	Upconverter
NDSatCom-KuLBand-Downconverter	2.07 200116	NDSatCom	Downconverter
NDSatCom-KuLBand-Upconverter	2.06 230715	NDSatCom	Upconverter
NDSatCom-RCU-RC11	1.00 251001		
NDSatCom-RCU-RC11OMa	1.03 251001		
NDSatCom-RCU-SNMP	2.10 230626	NDSatCom	Redundancy-Controller
NDSatCom-SkyWAN-1070	1.12 190725	NDSatCom	Modem
NDSatCom-SkyWAN-IDU	2.10 190725	NDSatCom	Modem
NDSatCom-SkyWAN	1.01 171214	NDSatCom	Modem
NDSatCom-Step-Track	1.05 171214	NDSatCom	Antenna-Controller
NDSatCom-TWTA-BC30	1.02 240314	NDSatCom	HPA TWTA
NDSatCom-TWTA-REST	1.03 251001	NDSatCom	HPA TWTA
NDSatCom-TWTA-SNMP	2.13 240314	NDSatCom	HPA TWTA
NDSatCom-TWTA	2.12 241009	NDSatCom	HPA TWTA

Driver	Version	Vendor	Tags
Negate	1.02 171221	SatService	logical
Netgear-M4300-Monitor	1.02 200206	Generic	Interface-Monitor
Network-Vikinx-VD3232-Matrix	1.00 191011	Network	Switch-Matrix
Nevion-Sublime	1.01 171004	Nevion	Switch
Nevion-TVG420	1.04 190313	Nevion	IP-Gateway
Nevion-TVG425	2.01 190219	Nevion	IP-Gateway
Newtec-AZ110-S2-Only	1.01 241008	Newtec	Modulator
Newtec-AZ110	1.08 251006	Newtec	Modulator
Newtec-AZ200-1-To-1	1.01 171220	Newtec	Redundancy-Controller
Newtec-AZ410	1.03 220513	Newtec	Modem
Newtec-AZ910	1.07 171214	Newtec	Demodulator
Newtec-EL470	1.04 190725	Newtec	Modem
Newtec-FRC0710	1.01 230601	Newtec	Upconverter
Newtec-FRC0750	1.05 240926	Newtec	Block-Upconverter Combiner
Newtec-M6100	1.21 251006	Newtec	Modulator
Newtec-MCX7000-ASIIN	1.01 210203	Newtec	Gateway

Driver	*Version*	*Vendor*	*Tags*
Newtec-MCX7000-ASIOUT	1.00 210203	Newtec	Gateway
Newtec-MCX7000-Base	1.00 191029	Newtec	Gateway
Newtec-MCX7000-BISS-Descr	1.01 210617	Newtec	Gateway
Newtec-MCX7000-Demodulator	1.06 250702	Newtec	Demodulator
Newtec-MCX7000-IPIN	1.03 210617	Newtec	Gateway
Newtec-MCX7000-IPOUT	1.03 210617	Newtec	Gateway
Newtec-MCX7000-Modulator	1.00 250704	Newtec	Modulator
Newtec-MCX7000-Protection	1.00 191028	Newtec	Gateway
Newtec-MDM6100	1.03 230105	Newtec	Modulator
Newtec-NTC2080	1.13 210916	Newtec	Modulator
Newtec-NTC2142	1.02 240314	Newtec	Upconverter Downconverter
Newtec-NTC2180	1.04 210916	Newtec	Modulator
Newtec-NTC2277	1.17 251001	Newtec	Modulator
Newtec-PEP	1.03 171214	Newtec	Gateway
Newtec-USS0202	1.02 211214	Newtec	Redundancy-Controller
Newtec-USS0212	1.00 211214	Newtec	Redundancy-Controller

Driver	Version	Vendor	Tags
NMEA-GPS-Receiver	1.01 171214	Generic	GPS
Noise-Source-Measurement	1.01 240104	SatService	logical
Novella-Blockdownconverter	1.00 210518	Novella	Block-Downconverter
Novella-Blockupconverter	1.00 210518	Novella	Block-Upconverter
Novella-RU03	1.00 240807	Novella	Redundancy-Controller
Novella-RU04	1.03 240913	Novella	Redundancy-Controller
Novella-Tracking-Receiver	1.03 180223	Novella	Tracking-Receiver
NovelSat-NS2000	1.07 240516	Novelsat	Demodulator
Novotronik-GTA5200	1.03 171004	Novotronik	Coupler
Novotronik-GTS0118	1.01 171123	Novotronik	Switch-Controller
Novotronik-GTS0752	1.01 171123	Novotronik	Switch-Controller
Novotronik-GTS0752A	1.01 171123	Novotronik	Switch-Controller
Novotronik-GTS4600	1.12 171123	Novotronik	Switch-Controller
Novotronik-GTS5022	1.00 200121	Novotronik	Switch-Controller
Novotronik-GTS5460	1.01 171121	Novotronik	Switch-Controller
Novotronik-GTS5852	1.01 171123	Novotronik	Switch-Controller

Driver	Version	Vendor	Tags
Novotronik-GTS6300	1.05 180119	Novotronik	LNA/LNB-Controller Redundancy-Controller
Novotronik-GTS6303	1.03 220728	Novotronik	LNA/LNB-Controller Redundancy-Controller
Novotronik-GTS6305	1.01 171123	Novotronik	LNA/LNB-Controller
Novotronik-GTS6821	1.01 171123	Novotronik	Switch-Controller
Novotronik-GTS6832	1.01 171123	Novotronik	Switch-Controller
Novotronik-GTV2455	1.02 171123	Novotronik	Amplifier
Novotronik-MAS3160	1.00 171228	Novotronik	Switch-Matrix
Novotronik-MAS3240	1.00 200204	Novotronik	Switch-Matrix
Novotronik-MAS3660	1.00 171228	Novotronik	Switch-Matrix
Novotronik-MAS3670	1.00 200214	Novotronik	Switch-Matrix
Novotronik-MAS4488	1.00 200617	Novotronik	Switch-Matrix
Novotronik-MAS5321	1.03 221028	Novotronik	Switch-Matrix
Novotronik-MAS5513	1.01 171123	Novotronik	Switch-Matrix
Novotronik-MAS5514	1.00 171004	Novotronik	Switch-Matrix
Novotronik-Matrix24	1.02 171123	Novotronik	Switch-Matrix
Novotronik-Matrix32	1.00 171123	Novotronik	Switch-Matrix

Driver	Version	Vendor	Tags
Novotronik-Matrix8	1.01 170926	Novotronik	Switch-Matrix
Novus-ND2310D	1.00 220225	Novus Power Products	Distribution-Amplifier
Novus-NR2110-SNMP	1.01 231205	Novus Power Products	Frequency-Reference
Novus-NR2110	1.02 170831	Novus Power Products	Frequency-Reference
NPC-ACU-550	1.01 250825	NPC Systems	Antenna-Controller
NTT-HVD6100	1.04 240314	NTT Electronics	IRD
NTT-HVE9100	1.03 240314	NTT Electronics	Encoder
NTT-HVE9210	1.14 240314	NTT Electronics	Encoder
Number-Select	1.01 171221	SatService	logical
Number-Set	1.01 171221	SatService	logical
Off-Switch	1.00 180115	SatService	logical
OnTarget	1.01 230817	SatService	logical
OpenGear-Frame	1.03 231114	OpenGear	Main-Frame
ORB-Step-Track	1.01 171221	NDSatCom	Antenna-Controller
Output	1.05 180628	SatService	logical
Paradise-RCP2-1100-SNMP	1.03 230526	Paradise Datacom	Redundancy-Controller

Driver	Version	Vendor	Tags
Paradise-RCP2-SNMP	1.02 171215	Paradise Datacom	Redundancy-Controller
Paradise-RCP2	1.02 211222	Paradise Datacom	Redundancy-Controller
Paradise-SSPA-2200AM	1.00 201005	Paradise Datacom	HPA SSPA
Paradise-SSPA-SNMP	2.07 190725	Paradise Datacom	HPA SSPA
Parameter-Mirror	1.02 250909	SatService	logical
Pass-Scheduler	1.05 230613	SatService	logical
Peak-Beacon-Receiver	1.03 231212	Peak Communications	Tracking-Receiver
Peak-Blockdownconverter	1.01 240314	Peak Communications	Downconverter
Peak-Blockupconverter	1.02 240314	Peak Communications	Blockupconverter
Peak-DBU200	1.00 190320	Peak Communications	Redundancy-Controller Block-Downconverter
Peak-DBU200H	1.01 190320	Peak Communications	Redundancy-Controller Block-Downconverter
Peak-Downconverter-7000	1.03 240314	Peak Communications	Downconverter
Peak-IBDH	1.01 240314	Peak Communications	Redundancy-Controller
Peak-P700	1.03 190725	Peak Communications	Upconverter Downconverter
Peak-UPC7002	1.09 240418	Peak Communications	Uplink-Power-Control
Peak-Upconverter-7000	1.03 240314	Peak Communications	Upconverter

Driver	Version	Vendor	Tags
Pico-EnviroMon-EL005	1.02 171215	Pico Technology	Temperature-Sensor
Pointing-Lock	1.03 171221	SatService	logical
Pointing-Robot	2.05 180105	SatService	logical
Polarization-Select	1.03 171221	SatService	logical
Polsat-ACTS	1.00 230228	Polsat	Antenna-Controller
PowerCalc-Transponder	1.02 241211	SatService	logical
PowerCalc	1.02 171221	SatService	logical
Profline-Plus1-AudioReceiver	1.02 200204	Profline	IRD Audio-Decoder
Promax-Sathunter	1.03 171215	Promax	Measurement
Protection-Switch-1-To-1-Dual	1.03 201119	SatService	logical
Protection-Switch-1-To-1	2.05 221217	SatService	logical
Protection-Switch-2-To-1	1.02 231027	SatService	logical
Protection-Switch-Asymmetric	1.01 190705	SatService	logical
Protection-Switch-N-To-1	4.07 241127	SatService	logical
Protection-Switch-N-To-M	1.03 250318	SatService	logical
Protection-Switch-NoSave-1-To-1	1.02 241217	SatService	logical

Driver	Version	Vendor	Tags
Proxy-Device	1.01 171221	SatService	logical
PTF-ptf1220A-SNMP			
Qbit-Q572-AudioReceiver	1.03 240314	Qbit	IRD Audio-Decoder
RadeusLab-ACU-8200	1.02 220615	Radeus Labs	Antenna-Controller
Radyne-DM240	1.05 240314	Radyne Comstream	Modulator
Radyne-DMD2050	1.04 190725	Radyne Comstream	Modem
Radyne-DVB3030	1.12 210916	Radyne Comstream	Modulator
Radyne-RCS11	1.02 171215	Radyne Comstream	Redundancy-Controller
Radyne-SFC1450	1.04 240314	Radyne Comstream	Upconverter
Raritan-EMX	1.00 190725	Raritan	Temperature-Sensor Measurement
Remote-Alarm	1.02 171221	SatService	logical
Replicator	3.00 221202	SatService	logical
Report-Generator	2.00 190410	SatService	logical
ResearchConcepts-RC1500	1.03 190321	Research Concepts	Antenna-Controller
ResearchConcepts-RC2000	1.14 181004	Research Concepts	Antenna-Controller
ResearchConcepts-RC2500	1.01 180502	Research Concepts	Antenna-Controller
ResearchConcepts-RC3000	1.04 180322	Research Concepts	Antenna-Controller

Driver	Version	Vendor	Tags
Reset-Timer	1.02 171221	SatService	logical
RFDesign-Amp	1.00 180131	RF Design	Amplifier
RFDesign-FLCR-ODA	1.00 240912	RF Design	Optical-Links
RFDesign-FLCR	1.00 240912	RF Design	Optical-Links
RFDesign-FlexLink-S7B	1.00 220204	RF Design	Switch-Matrix
RFDesign-FlexLink-S7Bc	1.02 220826	RF Design	Switch-Matrix
RFDesign-LSC	1.04 240314	RF Design	Splitter Combiner
RFDesign-QLink	1.00 240912	RF Design	Optical-Links
RFDesign-RedLink	1.04 180322	RF Design	Optical-Links
RFDesign-RLC	1.00 190227	RF Design	Combiner
RFDesign-RLS	1.00 190227	RF Design	Splitter
RFDesign-RSC	1.02 171215	RF Design	Redundancy-Controller
Romantis-UHP	1.03 190725	Romantis	Modem
RuS-FPH	1.00 250910	RS	Measurement Spectrum-Analyzer
RuS-FSC	1.00 250910	RS	Measurement Spectrum-Analyzer
RuS-FSH	2.00 250910	RS	Measurement Spectrum-Analyzer

Driver	Version	Vendor	Tags
RuS-FSL	2.00 250910	RS	Measurement Spectrum-Analyzer
RuS-FSV	2.00 250910	RS	Measurement Spectrum-Analyzer
RuS-PKU-SSPA	1.02 201026	Rohde&Schwarz	SSPA
RX-Channel-ActivationBased	2.01 251001	SatService	logical
RX-Channel-ByChannel	2.03 251001	SatService	logical
RX-Channel-BySatellite	2.03 251126	SatService	logical
RX-Channel	1.00 201021	SatService	logical
RX-Channel	1.15 230719	SatService	logical
RX-ChannelSelect-CDB-DBonly	2.08 201019	SatService	logical
RX-ChannelSelect-CDB	2.09 180129	SatService	logical
RX-ChannelSelect	2.07 171221	SatService	logical
RX-LineSettings-CDB-DBonly	1.06 171221	SatService	logical
RX-LineSettings-CDB	1.05 171221	SatService	logical
S3Satcom-ACU	1.02 200518	S3 Satcom	Antenna-Controller
Sampler	1.01 171221	SatService	logical
Satellink-1to2-Redundant-LNA	1.02 171215	Satellink	LNA/LNB-Controller Redundancy-Controller

Driver	Version	Vendor	Tags
SatelliteSystems-3430-BTR-L	1.01 240105	Satellite Systems	Tracking-Receiver
SatService-ACU-IDU	1.02 171215	SatService	Antenna-Controller
SatService-ACU-ODM-Serial	1.06 230718	SatService	Antenna-Controller
SatService-ACU-ODM	3.05 230719	SatService	Antenna-Controller
SatService-ACU-Prediction	1.02 210621	SatService	logical
SatService-ACU2-Serial	2.06 230920	SatService	Antenna-Controller
SatService-ACU2	2.05 240423	SatService	Antenna-Controller
SatService-Beacon-Receiver	4.02 230911	SatService	Tracking-Receiver
SatService-Deice	1.04 191113	SatService	Deice-Controller
SatService-IO-FEP-Protection-1-To-1	2.04 201119	SatService	Redundancy-Controller
SatService-IO-FEP-Protection-2-To-1	1.04 241211	SatService	Redundancy-Controller
SatService-IO-FEP2	2.01 220823	SatService	I/O-Controller Redundancy-Controller Switch-Controller
SatService-IO-FEP	2.09 171026	SatService	I/O-Controller Redundancy-Controller Switch-Controller
SatService-LBRX-1MT	1.02 171215	SatService	Tracking-Receiver
SatService-LFRXTX	1.72 250725	SatService	Optical-Links

Driver	*Version*	*Vendor*	*Tags*
SatService-Matrix	2.03 180424	SatService	Switch-Matrix
SatService-Power-Sensor	1.01 171026	SatService	Powersensor Measurement
SatService-SatID	1.01 100806		
SatService-SMU	1.01 161026	SatService	Main-Frame Distributor Redundancy-Controller Switch-Controller
SatService-Temp-232	1.01 171026	SatService	Temperature-Sensor
SCC-Device	1.00 050509		
Scopus-E1X00-SNMP	1.07 240314	Scopus	Encoder Modulator
Scopus-IRD26XX	1.03 171215	Scopus	IRD
Scopus-IRD28XX	1.07 171215	Scopus	IRD
Scopus-IRD2900-SNMP	3.01 171215	Scopus	IRD
Scopus-UE9000-SNMP	3.03 210916	Scopus	Encoder Modulator
Selector-Two-Way	1.00 230210	SatService	logical
Selector	1.03 201210	SatService	logical
Sencore-Afiniti-Encoder	1.01 251124	Sencore	Encoder
Sencore-AG4400	1.05 171215	Sencore	IRD
Sencore-MIP6210	1.03 240314	Sencore	Recorder Player

Driver	Version	Vendor	Tags
Sencore-MRD3187-DVBS2	1.04 171215	Sencore	IRD
Sencore-MRD3187-QUAD	2.01 230615	Sencore	IRD
Sencore-MRD5800	1.01 240314	Sencore	IRD
Sencore-MRD7000	1.14 250708	Sencore	IRD
ServerTech-PDU-Sentry-3	1.01 180925	ServerTech	Power-Module
ServerTech-PDU-Sentry-4	1.07 230912	ServerTech	Power-Module
SES-Attn-Offset	1.01 171221	SatService	logical
SES-DAKS	1.02 251028	SatService	logical
SES-Level-GET	1.02 171221	SatService	logical
SES-Level-SET	1.02 171221	SatService	logical
SES-Power-Control-Box	1.02 250618	SatService	logical
SES-Sysinfo	1.06 190521	SatService	logical
Show-Orbit	1.01 171221	SatService	logical
Sicilsat-S00761	1.00 250716	Sicilsat	Antenna-Controller
Site-Switch	1.02 171221	SatService	logical
SNG-Pointing	1.02 171221	SatService	logical

Driver	Version	Vendor	Tags
SNMP-Driver-Template	1.00 120815		
Spacepath-CM2-TWTA	1.02 250305	Spacepath	HPA TWTA
Spacepath-Tango-TWTA	1.00 240822	Spacepath	HPA TWTA
Spacepath-TWTA-Serial	1.02 210125	Spacepath	HPA TWTA
Spacepath-TWTA	1.07 240314	Spacepath	HPA TWTA
Speciality-BDC-Controller	1.02 171220	Speciality Microwave	Redundancy-Controller LNA/LNB-Controller
Speciality-Controller-1926-100	1.02 240314	Speciality Microwave	Redundancy-Controller
Speciality-Controller-9753-500	1.01 240314	Speciality Microwave	Redundancy-Controller
Spectrum-QAccess	1.04 200924	SatService	logical
SPX-Etsiline-CommPact-v101	1.02 171010	SPX	Dehydrator
SPX-Etsiline-CommPact	2.00 171024	SPX	Dehydrator
SRG-ChannelSelect	2.02 171221	SatService	logical
SSE-ASat	1.03 240314	SSE	Transceiver
SSE-KStar	1.06 240314	SSE	Transceiver
Store-Float	1.00 231117	SatService	logical
Stream-KeySelect	1.01 251001	SatService	logical

Driver	Version	Vendor	Tags
SVS-SWC100	1.01 251002		
Switch-Cross-Link	1.01 171221	SatService	logical
Symmetric-Site-Switch	1.00 250730	SatService	logical
Symmetricom-4036B	1.02 190227	Symmetricom	Distribution-Amplifier
Symmetricom-4059B	1.00 190227	Symmetricom	Distribution-Amplifier
Symmetricom-9611B	1.02 171215	Symmetricom	Distribution-Amplifier
Symmetricom-S650	1.00 240418	Symmetricom	GPS Frequency-Reference
Symmetricom-XLI	1.01 240314	Symmetricom	GPS Frequency-Reference
Sync-Switch	1.02 171221	SatService	logical
Synology-DSM	1.01 200624	Synology	Server
Sysinfo	3.15 250728	SatService	Server
T21-9261-Decoder	1.05 250630	Vitec T21	IRD
T21-9261-Encoder	1.01 250403	Vitec T21	Encoder
Tadiran-NTC2077	1.03 210916	Tadiran	Modulator
Tadiran-Scopus-E1000	1.05 240314	Tadiran	Encoder
Tandberg-Alteia	1.03 171218	Tandberg	IRD

Driver	Version	Vendor	Tags
Tandberg-E57XX-SNMP	4.03 210916	Tandberg	Encoder Modulator
Tandberg-RX1290-SNMP-V3.9	3.01 171218	Tandberg	IRD
Tandberg-RX1290-SNMP	4.06 171218	Tandberg	IRD
Tandberg-RX8200-SNMP	1.25 171218		
Tandberg-RX8200-SNMP	1.34 251006	Tandberg	IRD
Tandberg-RX8200-TriBand-SNMP	1.23 240314	Tandberg	IRD
Tandberg-SM62xx	1.00 230209	Tandberg	Modulator
Tandberg-TT1260-SNMP-PREV5	1.04 171218	Tandberg	IRD
Tandberg-TT1260-SNMP	4.05 180206	Tandberg	IRD
Tandberg-TT1260	1.02 171218	Tandberg	IRD
Technotrans-HC-Vertex	1.16 200730	Technotrans	Air-Conditioning
Teledyne-QFlex-400	1.01 250715	Teledyne Paradise	Modem
Teledyne-QMultiFlex	1.01 250715	Teledyne Paradise	Modem
Temperature-Converter	1.03 220824	SatService	logical
Teracue-TAC	1.01 200912	Teracue	Switch-Matrix
Terrasat-IBUC-SNMP	4.12 251124	Terrasat	Block-Upconverter

Driver	Version	Vendor	Tags
Terrasat-IBUC	1.06 190725	Terrasat	Block-Upconverter
Terrasat-RX-Protection	1.02 240314	Terrasat	Redundancy-Controller
Threshold	1.01 171221	SatService	logical
Time-Test	1.01 220920	SatService	logical
TimeLink-Distributor-TMD	1.00 200515	TimeLink	Frequency-Reference Distributor
TimeLink-Distributor	1.01 171218	TimeLink	Frequency-Reference Distributor
TimeLink-GNSS-Receiver-TMG3400	1.01 200629	TimeLink	GPS Frequency-Reference
TimeLink-GPS-Receiver	1.01 171218	TimeLink	GPS Frequency-Reference
TimeLink-TMG3420	1.01 251113	TimeLink	GPS Frequency-Reference
Timetech-Distributor	1.03 171218	Timetech	Frequency-Reference Distributor
Timetech-GPS-Receiver	1.03 171218	TimeLink	GPS Frequency-Reference
TLE-Updater	1.00 210624	SatService	logical
Toggle-Switch	1.01 171221	SatService	logical
Transmit-Chain-Preset	2.01 171221	SatService	logical
Triple-Translator	1.01 171221	SatService	logical
TrippLite-PDU	1.02 240314	Tripp-Lite	Power-Module

Driver	*Version*	*Vendor*	*Tags*
Tvips-TNS541-SNMP	1.02 171218	TVIPS	Measurement
TX-Channel-ByChannel	2.02 251001	SatService	logical
TX-Channel-BySatellite	2.01 251001	SatService	logical
TX-Channel	2.01 211014	SatService	logical
TX-ChannelSelect-CDB	2.07 171221	SatService	logical
TX-ChannelSelect	2.01 171221	SatService	logical
TX-LineSettings-CDB	1.05 171221	SatService	logical
TX-OFF-Timer	2.03 171221	SatService	logical
UDP-Sender	1.00 190708	SatService	logical
Uplink-Power-Control-2	1.02 171222	SatService	Uplink-Power-Control
Uplink-Power-Control	1.12 241126	SatService	Uplink-Power-Control
V4L-DVBS2-Demod	1.00 120116	TurboSight	Demodulator
Vaisala-WXT	1.07 200916	Vaisala	Weather
Vertex-7134-Step-Track	2.05 200219	Vertex	Antenna-Controller
Vertex-7200-Step-Track-DeltaPol	2.16 201208	Vertex	Antenna-Controller
Vertex-7200-Step-Track-Prec	2.16 201208	Vertex	Antenna-Controller

Driver	Version	Vendor	Tags
Vertex-7200-Step-Track	2.18 230817	Vertex	Antenna-Controller
Vertex-ACU5000	1.12 250723	Vertex Antennentechnik	Antenna-Controller
Vertex-ACU8100	1.09 180716	Vertex Antennentechnik	Antenna-Controller
Vertex-Digital-Tracking-Receiver	1.07 200412	Vertex	Tracking-Receiver
Vertex-Downconverter-legacy	1.06 251107	Vertex	Downconverter
Vertex-RSC-1100	1.01 171124	Vertex	LNA/LNB-Controller Redundancy-Controller
Vertex-RSC-1200	1.07 240314	Vertex	LNA/LNB-Controller Redundancy-Controller
Vertex-SCR-Downconverter-SNMP-legacy	1.03 251107	Vertex	Downconverter
Vertex-SCR-Downconverter-SNMP	1.00 140214		
Vertex-SCR-Downconverter	1.02 200116	Vertex	Downconverter
Vertex-SCR-Upconverter-SNMP-legacy	1.03 230323	Vertex	Upconverter
Vertex-SCR-Upconverter	1.02 190725	Vertex	Upconverter
Vertex-SSPA	2.05 240314	Vertex	Block-Upconverter HPA SSPA
Vertex-TWTA-2100TK	1.06 240314	Vertex	TWTA HPA
Vertex-Upconverter-legacy	1.04 230323	Vertex	Upconverter

Driver	Version	Vendor	Tags
ViaLite-HRA-Dual-EDFA	1.02 241126	ViaLite	Optical-Links Amplifier
ViaLite-HRA-EDFA	1.05 250331	ViaLite	Optical-Links Amplifier
ViaLite-HRS-Switch-4way	1.00 250403	ViaLite	Optical-Links Switch-Controller
ViaLite-HRS-Switch	1.01 200110	ViaLite	Optical-Links Switch-Controller
ViaLiteHD-FO	1.06 250618	ViaLite	Optical-Links
ViaSat-4100-ACU	1.09 240123	ViaSat	Antenna-Controller
ViaSat-EBEM	1.04 250214	ViaSat	Modem
VioIP-End-Point	1.04 190128	SatService	logical
Viprinet-Hub-Redundany-HTTPS	1.00 171130	Viprinet	Router IP-Gateway
Viprinet-HUB-SNMP	1.02 180119	Viprinet	Router IP-Gateway
Viprinet-HUB	1.02 170914		
Viprinet-Router-HTTPS	1.06 160304		
Viprinet-Router-SNMP	1.00 150209		
Viprinet-Router	1.02 160626		
Viprinet-Rugged-Router-HTTPS	1.07 190605	Viprinet	Router IP-Gateway
Viprinet-Rugged-Router-SNMP	1.04 190328	Viprinet	Router IP-Gateway

Driver	Version	Vendor	Tags
Viprinet-Tunnel-SNMP	1.04 180119	Viprinet	Router IP-Gateway
Viprinet-Tunnel	1.02 150626		
Virtual-Matrix-128x128	1.01 171221	SatService	logical
Virtual-Matrix-128x32	1.03 171221	SatService	logical
Vitec-MGW-Encoder	1.01 250213	Vitec	Encoder Gateway
Vocality-FourWire	1.02 171220	Vocality	Intercom
Walton-DDC-Deice	1.00 210825	Walton	Deice-Controller
Walton-Deice	1.05 220923	Walton	Deice-Controller
Wavestream-BUC-SNMP	1.03 240314	Wavestream	Block-Upconverter
Wavestream-BUC	1.02 190725	Wavestream	Block-Upconverter
Wegener-UNITY44XX	1.04 171218	Wegener	IRD
WG-Switch	2.01 171221	SatService	logical
WG-Switch	2.03 171221	SatService	logical
Wilmers-blueberryCOMPACT	1.05 180223	Wilmers	Weather
WISI-Chameleon-DVBC	1.01 240314	Wisi	IRD
Work-Amplifier	1.02 171114	Work Microwave	Amplifier

Driver	Version	Vendor	Tags
Work-AT61-Modulator	1.03 240924	Work Microwave	Modulator
Work-DualSBD	1.09 201105	Work Microwave	Block-Downconverter
Work-DualSBU	1.11 240119	Work Microwave	Block-Upconverter
Work-DualSSBD	1.03 200116	Work Microwave	Block-Downconverter
Work-DualSSBU	1.04 190910	Work Microwave	Block-Upconverter
Work-DualVSCD	1.13 210721	Work Microwave	Block-Downconverter
Work-DualVSCU	1.12 210721	Work Microwave	Block-Upconverter
Work-MBU-MiniBuc	2.02 190725	Work Microwave	Block-Upconverter
Work-QuadSBD	1.04 200116	Work Microwave	Block-Downconverter
Work-QuadSBU	1.06 190910	Work Microwave	Block-Upconverter
Work-RSC	1.04 210630	Work Microwave	Redundancy-Controller
Work-RSCM-Dual	1.01 211222	Work Microwave	Redundancy-Controller
Work-RSCM-Quad	1.00 230607	Work Microwave	Redundancy-Controller
Work-RSCM-Triple	1.02 220417	Work Microwave	Redundancy-Controller
Work-RSCM	1.11 211222	Work Microwave	Redundancy-Controller
Work-SBD	1.03 200116	Work Microwave	Block-Downconverter

Driver	Version	Vendor	Tags
Work-SBU	1.05 190910	Work Microwave	Block-Upconverter
Work-SCD	1.04 200116	Work Microwave	Downconverter
Work-SCU	1.05 200219	Work Microwave	Upconverter
Work-SDM2-Modulator	1.08 190725	Work Microwave	Modulator
Work-SDM2-MT6-Modulator	1.03 190725	Work Microwave	Modulator
Work-SDM2-MT6-Modulator	1.03 190725	Work Microwave	Modulator
Work-SM2CU-Modulator	1.04 210916	Work Microwave	Modulator
Work-SSBD	2.02 200116	Work Microwave	Block-Downconverter
Work-SSBU	2.03 230901	Work Microwave	Block-Upconverter
Work-STLT2-KuK	1.02 181016	Work Microwave	Testloop-Translator
Work-TLT-LO	1.05 210721	Work Microwave	Testloop-Translator
Work-TLT	2.07 210721	Work Microwave	Testloop-Translator
Work-TripleVSCD	1.02 210721	Work Microwave	Block-Downconverter
Work-TriSBD	1.00 211112	Work Microwave	Block-Downconverter
Work-TriSBU	1.01 220228	Work Microwave	Block-Upconverter
Work-VCR	1.00 181015	Work Microwave	Controller

Driver	Version	Vendor	Tags
Work-VSCD	1.09 210721	Work Microwave	Block-Downconverter
Work-VSCSR	1.04 190725	Work Microwave	Frequency-Reference
Work-VSCU	1.13 221123	Work Microwave	Block-Downconverter
Work-VSM2CU-Modulator-Delay	1.15 221025	Work Microwave	Modulator
Work-VSM2CU-Modulator	1.13 220113	Work Microwave	Modulator
Work-VSSBD	2.05 210721	Work Microwave	Block-Downconverter
Work-VSSBU	2.04 210721	Work Microwave	Block-Upconverter
Xicom-KHPA	1.06 240314	Xicom	HPA Klystron
Xicom-TWTA-XT	2.04 240314	Xicom	HPA TWTA
Xicom-TWTA-XTD	1.12 240314	Xicom	HPA TWTA
Xicom-TWTA-XTDL	1.21 240314	Xicom	HPA TWTA UPC
Xicom-TWTA-XTRT	1.18 240314	Xicom	HPA TWTA
Xicom-TWTA	2.09 240314	Xicom	TWTA HPA
XMW-Blockupconverter	1.00 240606	XMW	Blockupconverter
Xytronix-X600	1.03 230223	Xytronix	I/O-Controller Switch-Controller
Zodiac-Cortex-ACU	2.02 240419	Zodiac	Antenna-Controller

Driver	Version	Vendor	Tags
Zodiac-Cortex-CRT-IFM	1.10 240527	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT-IFR	1.08 240527	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT-PRJ	1.05 230323	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT-TCU	1.09 240313	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT-TMS	1.05 240313	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT-TMU	1.07 240821	Zodiac	Modulator Demodulator
Zodiac-Cortex-CRT	1.06 240313	Zodiac	Modulator Demodulator
Zodiac-Cortex-DTR-Cal	1.03 171220	Zodiac	Tracking-Receiver
Zodiac-Cortex-DTR-Path	1.04 171220	Zodiac	Tracking-Receiver
Zodiac-Cortex-DTR-Q	1.05 221220	Zodiac	Tracking-Receiver
Zodiac-Cortex-DTR	1.07 170914	Zodiac	Tracking-Receiver
Zodiac-Cortex-DTR	1.08 190312	Zodiac	Tracking-Receiver
Zodiac-Cortex-DTR	1.09 200917	Zodiac	Tracking-Receiver
Zodiac-Cortex-Reset	1.00 250224		

9.2 Device Drivers By Vendor

Vendor	Driver	Version	Tags

Vendor	Driver	Version	Tags
Acorde Technologies	Acorde-ACRS-IDU	1.00 250916	Redundancy-Controller LNA/LNB-Controller
	Acorde-BUC-ACCVT	1.06 240807	Block-Upconverter
ACTi	ACTi-ACM1231-Webcam	1.03 171212	Camera
Actia-Sodielec	Actia-Sodielec-DNT-HPA	1.08 240314	HPA TWTA
	Actia-Sodielec-DSS1300	1.02 171212	I/O-Controller
Adtec	Adtec-Afiniti-Decoder	1.01 220423	IRD
	Adtec-Afinity-Encoder	1.03 230202	Encoder
	Adtec-EN81	1.11 210916	Encoder Modulator
	Adtec-RD60	1.08 171212	IRD
Advantech	Advantech-ADAM5K	1.02 171212	I/O-Controller
	Advantech-ADAM6K	1.01 171212	I/O-Controller
Advantech Wireless	Advantech-Genesis	1.01 241212	HPA SSPA
	Advantech-SSPB-GCP	1.01 240314	HPA SSPA
	Advantech-SSPB-MCP	1.01 240314	HPA SSPA
	Advantech-SSPB-SNMP	1.06 250715	HPA SSPA
	Advantech-SSPB	1.09 190725	HPA SSPA

Vendor	Driver	Version	Tags
Agilent	Agilent-Spectrum-Analyzer	1.04 171220	Spectrum-Analyzer Measurement
Agilis	Agilis-BUC	1.01 190725	Block-Upconverter
Altec	Altec-Puregas-P550	1.01 240314	Dehydrator
Altec Air	Altec-Puregas-P210	1.00 240514	Dehydrator
Anacom	Anacom-KuBand-Transceiver	1.05 240314	Transceiver
	Anacom-Protection-Switch	1.01 171212	Redundancy-Controller
Andrew	Andrew-APC100	1.04 171220	Antenna-Controller
Antech	Antech-Deice	1.01 171220	Deice-Controller
Antwerpspace	AntwerpSpace-G3-Demod	1.09 230310	Demodulator
	AntwerpSpace-G3-Mod	1.08 230310	Modulator
	AntwerpSpace-IBB-Modem	1.15 240314	Modem
AP-Kieffer	APKieffer-TWTA-AC	1.02 240314	Air-Conditioning
APC	APC-ATS	1.02 240314	Power-Module
	APC-RPDU	1.03 190130	Power-Module
Apollo	Apollo-Switch	1.02 171212	Redundancy-Controller
ASCSignal	ASCSignal-NGC-ACU	1.07 190725	Antenna-Controller

Vendor	Driver	Version	Tags
	ASCSignal-NGC-BDC	1.02 200309	Block-Downconverter
	ASCSignal-NGC-Deice	1.00 190725	Deice-Controller
	ASCSignal-NGC-Hub-AC	1.00 190725	Air-Conditioning
	ASCSignal-NGC-RedundancyModule	2.01 240314	Redundancy-Controller LNA/LNB-Controller
	ASCSignal-NGC-Temperature	1.00 190725	Temperature-Sensor
	ASCSignal-NGC-Weather	1.00 190725	Weather
ASTRO Strobel Kommunikationssysteme GmbH	Astro-EdgeStreamer	1.05 250708	IRD
Ateme	Ateme-DR5000	3.01 250606	IRD
	Ateme-DR8400	1.10 240314	IRD
	Ateme-Kyrion-CM5000	1.25 250526	Encoder Modulator
	Ateme-TitanEdge-Bridge	1.02 251203	Video-Converter
	Ateme-TitanEdge-Encoder	1.11 251203	Encoder
	Ateme-TitanEdge-Gateway	1.11 251203	Gateway
	Ateme-TitanEdge-IRD	2.22 251126	IRD
Aten	Aten-VGA-Splitter	1.01 171212	Graphics
Atlantic Satellite Corporation	AtlanticSat-ASC300	1.02 230911	Tracking-Receiver

Vendor	Driver	Version	Tags
Atlantic-Microwave	AtlanTecRF-TLT	1.02 220929	Testloop-Translator
Automation-Direct	AutomationDirect-Terminator	2.00 190312	I/O-Controller
AvL	AvL-ITC-ACU	1.07 251104	Antenna-Controller
AvL Technologies	AvL-TracStar	1.08 180323	Antenna-Controller
Barco	Barco-RE4220	1.03 240314	Encoder
Bard	Bard-MC4002-AC	1.00 231130	Air-Conditioning
Blankom	Blankom-DRP393-SNMP	1.07 240314	IRD
BMS-IOM	BMS-IOM-AHU	1.03 240314	Measurement
	BMS-IOM-Generator	1.01 1800810	Generator
	BMS-IOM-Powermeter	1.01 180810	Powersensor
	BMS-IOM-Socomec-Powermeter	1.01 180810	Powersensor
	BMS-IOM-Trend-DI	1.01 180810	I/O-Controller
	BMS-UPS	1.01 180813	UPS
Brainbox	Brainbox-EDxxx	1.01 240314	I/O-Controller
Calian	Calian-Decimator	1.06 240918	Spectrum-Analyzer Measurement
	Calian-MCI-ACU	1.05 250909	Antenna-Controller

Vendor	Driver	Version	Tags
CandM	CandM-Switch-30510	1.00 220928	Redundancy-Controller
	CandM-Switch-308901	1.00 220621	Redundancy-Controller
	CandM-Switch	1.02 210819	Redundancy-Controller
Chaparral Communications	Monterey-140	1.01 171214	Antenna-Controller
China Suncom	ChinaSuncom-9293S3-ACU	1.00 231025	Antenna-Controller
	ChinaSuncom-Beacon-Receiver-SC9800	1.00 231030	Tracking-Receiver
Cibred Sud	Cibred-LAB4	1.01 190515	Dehydrator
Cisco	Cisco-2800GSM-SNMP	1.03 171212	Router
	Cisco-2900-SNMP	1.04 171212	Router
	Cisco-Monitor	1.02 171212	Router
	Cisco-Router	1.01 171212	Router
Cisco Synamedia	Cisco-D9800-IRD	1.04 210608	IRD
Codan	Codan-BUC-7586L	1.01 230323	Block-Upconverter
	Codan-LBUC-6x00	2.04 240314	Block-Upconverter
	Codan-Transceiver	1.04 240314	Block-Upconverter Block-Downconverter
Codan CPI	Codan-7586L	1.010 230323	Redundancy-Controller

Vendor	Driver	Version	Tags
Comstream	Comstream-ABR200	1.01 171212	IRD
	Comstream-ABR202	1.05 171212	IRD
	Comstream-CM601A-SCPC	1.06 190725	Modem
	Comstream-CM701-MC	1.02 190725	Modem
	Comstream-CM701	1.02 190725	Modem
	Comstream-CX101	0.01 171212	Redundancy-Controller
	Comstream-CX801	1.02 171212	Redundancy-Controller
	Comstream-DAC700-PRE	1.04 171212	Encoder Decoder
	Comstream-DAC700	1.03 171212	Encoder Decoder
	Comstream-Demodulator	1.01 171212	Demodulator
	Comstream-LoSpeed-Modem-SCPC	1.05 210916	Modem
	Comstream-Modem-Card-SCPC	1.03 210916	Modem
	Comstream-Modulator	1.03 210917	Modulator
Comtech	Comtech-CDD564	1.00	Modem
	Comtech-CDM570	1.06 200420	Modem
	Comtech-CDM600	1.53 240314	Modem

Vendor	Driver	Version	Tags
	Comtech-CDM760	1.00 241210	Modulator Demodulator
	Comtech-DT4500-SNMP	1.02 200116	Downconverter
	Comtech-DT4500	1.06 200116	Downconverter
	Comtech-LBC4000	1.00 250428	Upconverter Downconverter
	Comtech-LPOD-SNMP	1.05 240314	Block-Upconverter
	Comtech-MBT5000	1.02 240314	Block-Upconverter Block-Downconverter
	Comtech-MBT5000A	1.00 250515	Block-Upconverter Block-Downconverter
	Comtech-UT4500-SNMP	1.02 190725	Upconverter
	Comtech-UT4500	1.07 250724	Upconverter
Comtech CGC	CGC-ACU	1.18 240611	Antenna-Controller
ComWinTop	ComWinTop-THxxS	1.02 250910	Temperature-Sensor
CPI	CPI-350W-TWTA	2.05 240314	HPA TWTA
	CPI-CHPA	1.09 241009	TWTA HPA
	CPI-GaN-SSPA	1.01 240314	SSPA HPA
	CPI-KHPA-G3	2.11 240313	HPA Klystron
	CPI-KHPA-G4	2.15 250702	HPA Klystron

Vendor	Driver	Version	Tags
	CPI-LPA	2.06 240313	HPA TWTA
	CPI-Mini-ODU	1.27 241119	TWTA HPA
	CPI-SLPA	1.11 250610	HPA TWTA
	CPI-SRCU-BCIP	1.01 240314	Redundancy-Controller
	CPI-SRCU	1.00 190326	Redundancy-Controller
	CPI-Switch	1.03 210819	Redundancy-Controller
	CPI-TouchPower	1.13 240917	TWTA HPA
	CPI-TWTA-Ka	2.07 240314	HPA TWTA
	CPI-TWTA	2.06 240314	HPA TWTA
	CPI-VZU-TWTA	1.04 240314	HPA TWTA
Cross Technology	Cross-Frequency-Source	1.01 171213	Frequency-Reference
	Cross-UpDownconverter	1.03 190725	Upconverter Downconverter
Dantherm	Dantherm-ComboCooling1000	1.01 250610	Air-Conditioning
	Dantherm-TKS3000A	1.02 171213	Air-Conditioning
Dektec	Dektec-DTA2137	3.07 200114	Demodulator
	Dektec-DTA2137	1.07 130408	Demodulator

Vendor	Driver	Version	Tags
DEV Systemtechnik	DEV-4111-DA	1.01 231025	Distribution-Amplifier
	DEV-Distribution-2190	1.02 171006	Distribution-Amplifier
	DEV-FO-RX	1.00 201019	Optical-Links
	DEV-FO-TX	1.00 201019	Optical-Links
	DEV-Frame	1.00 201019	Main-Frame
	DEV-Matrix-1986	1.06 240314	Switch-Matrix
	DEV-Matrix-1996	1.00 250827	Switch-Matrix
	DEV-Switch-2-To-1	1.00 201202	Switch
DoubleD Electronics	DoubleD-DDA89	1.00 251114	Redundancy-Controller
E2V	E2V-TWTA	1.06 200728	HPA TWTA
ECU Corporation	ECU-J0475	1.00 230810	Air-Conditioning
EF-Data	EF-Data-SDM300	1.02 190725	Modem
Emerson Liebert	Emerson-AC-HPM	1.01 171213	Air-Conditioning
Endrun Technologies	Endrun-FDC3300	1.02 170926	Distributor
Enensys	Enensys-Vyper-Modulator	1.05 251104	Modulator
Environmental Technology	ET-ADH-2A	1.02 210519	Dehydrator

Vendor	Driver	Version	Tags
	ETI-ADH-NETCOM	1.07 231220	Dehydrator
	ETI-ASM-1-Manifold	1.01 231103	Dehydrator
Envivo	Envivio-4Caster-B3	1.04 171213	Encoder
	Envivio-4View-D1	1.02 171213	IRD
Ericsson	Ericsson-AVP	1.06 250224	Encoder Modulator
	Ericsson-Voyager-II-v9.5	3.19 210916	Encoder Modulator
	Ericsson-Voyager-II	4.03 210916	Encoder Modulator
ETL Systems	ETL-2773-XXYY	1.01 240314	LNA/LNB-Controller
	ETL-Alto-Amp	1.04 180814	LNA/LNB-Controller Amplifier
	ETL-Alto-DA	1.02 240314	Distribution-Amplifier
	ETL-Dextra-Combiner	1.01 240314	Combiner
	ETL-Dextra-DistAmp	1.07 190717	Distribution-Amplifier
	ETL-Genus-100MHz	1.00 251020	Frequency-Reference
	ETL-Genus-10MHz	1.00 240315	Frequency-Reference
	ETL-Genus-ALT	1.00 230526	Splitter
	ETL-Genus-COM402	1.02 221108	Combiner

Vendor	Driver	Version	Tags
	ETL-Genus-DA-22512	1.07 240103	Distribution-Amplifier
	ETL-Genus-DIV401	1.02 210521	Splitter
	ETL-Genus-Downconverter	1.01 251020	Downconverter
	ETL-Genus-Frame	1.05 250217	Main-Frame
	ETL-Genus-PRN	1.01 250925	LNA/LNB-Controller
	ETL-Genus-SRY-RX	1.00 240826	Optical-Links
	ETL-Genus-SRY-TX	1.01 250606	Optical-Links
	ETL-Genus-Upconverter	1.01 251020	Upconverter
	ETL-Genus-VTR-101-Matrix	1.02 240428	Switch-Matrix
	ETL-Genus-VTRC-101-Matrix	1.02 240428	Switch-Matrix
	ETL-Hurricane-Matrix	1.01 190403	Switch-Matrix
	ETL-LS-Switch	1.09 241213	Switch-Controller
	ETL-Modular-Frame-26128	1.04 180429	Main-Frame
	ETL-NGM100-Matrix	2.07 250527	Switch-Matrix
	ETL-NGM100C-Matrix	1.02 250519	Switch-Matrix
	ETL-NGM21C-Matrix	1.12 250519	Switch-Matrix

Vendor	Driver	Version	Tags
	ETL-NGM30-Matrix	1.13 250519	Switch-Matrix
	ETL-NSN100-Matrix	4.02 250527	Switch-Matrix
	ETL-Piranha	1.01 240314	LNA/LNB-Controller
	ETL-POE-Switch	1.01 240314	Switch-Controller
	ETL-SHF-Switch-2to2	1.00 250428	Switch-Controller
	ETL-SHF-Switch	1.02 171220	Switch-Controller
	ETL-Splitter-DIV409	1.03 171220	Splitter
	ETL-Splitter-DIV429	1.00 190920	Splitter
	ETL-Stingray-FO-Base-Serial	1.00 201020	Distribution-Amplifier
	ETL-Stingray-FO-Base	1.00 200922	Distribution-Amplifier
	ETL-Stingray-FO-RX-Serial	1.00 201020	Optical-Links
	ETL-Stingray-FO-RX	1.01 211210	Optical-Links
	ETL-Stingray-FO-TX-Serial	1.00 201020	Optical-Links
	ETL-Stingray-FO-TX	1.01 211210	Optical-Links
	ETL-SwitchUnit-23258	1.03 180814	Switch-Controller
	ETL-Volcano-Matrix	1.03 240314	Switch-Matrix

Vendor	Driver	Version	Tags
	ETL-VTR70-Matrix	1.04 251006	Switch-Matrix
	ETL-VTR70C-Matrix	1.03 251021	Switch-Matrix
	ETL-VTX100-Matrix	1.03 240702	Switch-Matrix
Evertz	Evertz-7700-Frame	1.01 250901	Main-Frame
	Evertz-7703-BPX	1.02 190716	Optical-Links
	Evertz-7707-ET	1.00 181206	Optical-Links
	Evertz-7707-LR	1.01 190321	Optical-Links
	Evertz-7707-LT	1.01 190321	Optical-Links
	Evertz-7708-LR	1.01 190321	Optical-Links
	Evertz-7708-LT	1.00 190321	Optical-Links
	Evertz-7800-Frame	1.02 240314	Main-Frame
	Evertz-7881-IRD	1.02 240314	IRD
	Evertz-EMR-Matrix	1.00 250204	Switch-Matrix
Fanuc	Fanuc-PLC9030	1.03 171213	I/O-Controller
FeiZyfer	FeiZyfer-GPS-Receiver	1.03 250703	GPS Frequency-Reference
Foxcom	Foxcom-SatLight-Platinum	1.12 190515	Optical-Links

Vendor	Driver	Version	Tags
GD Satcom	GDSatcom-ACU930	1.01 250327	Antenna-Controller
	GDSatcom-ACU940	1.05 250521	Antenna-Controller
	GDSatcom-ACU950-MP	1.03 250407	Antenna-Controller
	GDSatcom-ACU950	1.14 250407	Antenna-Controller
	GDSatcom-ACU970	1.06 250407	Antenna-Controller
	GDSatcom-ACU980	1.01 250407	Antenna-Controller
	GDSatcom-DTR520	1.03 240916	Tracking-Receiver
	GDSatcom-RSC1200	2.00 220317	Redundancy-Controller LNA/LNB-Controller
GDSatcom	GDSatcom-SSPA-QuadMod	1.02 240314	SSPA
GDSatcom CPI	GDSatcom-Rack-SSPA	1.00 240314	SSPA
Generex	Generex-CS121-UPS-Adapter	1.01 171213	UPS
Generic	MIB2-Ethernet-Switch	1.01 240523	Interface-Monitor
	MIB2-Interface-Monitor	1.04 191010	Interface-Monitor
	MIB2-Storage-Monitor	1.01 250224	Storage-Monitor
	MIB2-UPS-Monitor	1.04 250428	UPS
	Netgear-M4300-Monitor	1.02 200206	Interface-Monitor

Vendor	*Driver*	*Version*	*Tags*
	NMEA-GPS-Receiver	1.01 171214	GPS
Geosync Microwave	Geosync-PS	1.00 181105	Power-Supply
	Geosync-RCU-R2R	1.01 210924	Redundancy-Controller
	Geosync-RCU	1.07 251030	LNA/LNB-Controller Redundancy-Controller
	Geosync-TLT	1.04 190725	Testloop-Translator
Gude	Gude-ExpertPowerControl	1.01 171213	Power-Module
Guntermann und Drunck	GundD-CompactCenter	1.01 171213	Graphics
Haivision	Haivision-Makito-X-Decoder	1.04 250910	Decoder
	Haivision-Makito-X	1.07 251112	Encoder
	Haivision-SRT-Gateway	1.07 251104	Gateway
Harris	Harris-Panacea	1.02 171213	Switch-Matrix
Hewlett-Packard	HP-Powermeter-Legacy	1.00 201021	Powersensor Measurement
	HP-Powermeter-SCPI	1.02 171220	Powersensor Measurement
	HP-Synthesizer-SCPI	1.03 220823	Synthesizer
Hilkom	Hilkom-BNova-SNMP	1.02 171213	IRD
Hiltron	Hiltron-ACU2	1.07 250514	Antenna-Controller

Vendor	Driver	Version	Tags
	Hiltron-Deice	1.03 180115	Deice-Controller
	Hiltron-HCS-Frame	1.00 180110	Main-Frame
	Hiltron-HCS-Power	1.01 171213	Optical-Links LNA/LNB-Controller
	Hiltron-HCS-PS	1.01 180117	Power-Supply
	Hiltron-HCS-SSPA	1.03 190725	HPA SSPA
	Hiltron-HCS-Switch	1.01 171213	Redundancy-Controller
	Hiltron-HCS-TWTA	1.06 200728	HPA TWTA
	Hiltron-Multi-Deice	1.00 201023	Deice-Controller
Hitec	Hitec-AFP-Deice	1.02 171213	Antenna-Controller
	Hitec-AFP-Deice	1.02 171213	Deice-Controller
	Hitec-HACU-01-NAOS	1.22 230427	Antenna-Controller
	Hitec-HACU-01-V3.02	1.00 250723	Antenna-Controller
	Hitec-HACU-01-V3	2.01 250923	Antenna-Controller
	Hitec-HACU-01	1.09 211027	Antenna-Controller
	Hitec-HAFP-00-NAOS1	1.04 230418	Deice-Controller
	Hitec-HAFP-00	1.04 190228	Deice-Controller

Vendor	Driver	Version	Tags
	Hitec-HSCU-00	1.05 251126	Deice-Controller
HW Group	HWGroup-Poseidon2	1.01 230207	Temperature-Sensor I/O-Controller
	HWGroup-STE	1.01 230209	Temperature-Sensor
	HWGroup-STE2	1.03 180223	Temperature-Sensor
iDirect	IDirect-5000	1.06 190725	Modem
Infratec	Infratec-PM	1.03 171213	Power-Module
IP Sens	IP-Sens-IPS2225P	1.00 240411	Temperature-Sensor
IRT Technologies	Irt-Pico-BUC-RCP	1.06 190725	Block-Upconverter
IRT Technologies Spacepath	Irt-Pico-BUC-RCP	1.01 220816	Block-Upconverter
Janitza	Janitza-UMG604	1.02 171220	Powersensor Measurement
KBR	KBR-Multimess-4F96	1.02 190212	Powersensor Measurement
Keysight	Keysight-FieldFox	1.1 250319	Measurement Spectrum-Analyzer
Leitch	Leitch-Panacea	1.01 171213	Switch-Matrix
LPT	LPT-3000RX4	1.0 250326	Measurement Spectrum-Analyzer
Lynx Technik	Lynx-DVD5420-ASI-Distr	1.02 171213	Distributor
	Lynx-DVD5820-ASI-Distr	1.02 171213	Distributor

Vendor	Driver	Version	Tags
	Lynx-DVD5820-Main	1.03 171213	Main-Frame
Maxtech	Maxtech-LRC-1200	1.04 220422	LNA/LNB-Controller Redundancy-Controller
MCL	MCL-MT-TWTA-CSP	1.10 250107	HPA TWTA
	MCL-MT-TWTA	2.05 200728	HPA TWTA
Mediakind	Mediakind-CE1	1.01 250326	Encoder
	Mediakind-RX1	1.02 250526	IRD
Meinberg	Meinberg-LantimeM300	1.02 221110	GPS Frequency-Reference
	Meinberg-SDU	1.00 240411	Distribution-Amplifier
Mesa	Mesa-WNT	1.02 180223	Weather
	Mesa-WSWD	1.02 180223	Weather
MiniCircuits	MiniCircuits-Switch-ZT	1.02 240314	Switch-Controller
Mirad Microwave	Mirad-VPC	1.03 171213	Switch-Controller Variable-Power-Combiner
Mission Microwave	MissionMicrowave-SSPA	0.06 240729	Block-Upconverter SSPA
Mitec	Mitec-ARK-RX	1.00 190710	LNA/LNB-Controller Redundancy-Controller
	Mitec-Switch	1.02 171213	Switch-Controller Variable-Power-Combiner

Vendor	Driver	Version	Tags
	Mitec-TX	1.01 240314	HPA SSPA
Miteq	Miteq-Downconverter-9900-SNMP	1.00 230126	Downconverter
	Miteq-Downconverter-9900	1.00 211008	Downconverter
	Miteq-DVM100	1.03 210916	Modulator
	Miteq-KBand-Upconverter	1.02 190725	Upconverter
	Miteq-KuBand-Upconverter	1.00 210512	Upconverter
	Miteq-LBand-Downconverter	1.03 200116	Downconverter
	Miteq-LBand-Upconverter	1.03 190725	Upconverter
	Miteq-NSU	1.04 250718	Upconverter Redundancy-Controller
	Miteq-TLT-LO	1.00 210517	Testloop-Translator
	Miteq-TLT	1.01 210928	Testloop-Translator
	Miteq-UPC	1.03 251230	Uplink-Power-Control
	Miteq-Upconverter-9900-SNMP	1.00 230126	Downconverter
Moxa	Moxa-ioLogik-1200	1.03 240314	I/O-Controller
	Moxa-ioLogik-2214	1.00 240402	I/O-Controller
	Moxa-ioLogik-4000	1.02 171124	I/O-Controller

Vendor	Driver	Version	Tags
	Moxa-ioLogik-4200	1.03 171124	I/O-Controller
	Moxa-ioLogik-Current	1.01 171124	I/O-Controller
	Moxa-ioLogik-Temperature	1.01 171124	I/O-Controller Temperature-Sensor
	Moxa-ioThinx-4510	1.00 210608	I/O-Controller
NAIS/Panasonic	NAIS-FP-8-Port-Switch	1.01 170926	I/O-Controller
	NAIS-FP-AD-Test	1.01 171214	I/O-Controller
	NAIS-FP-AD4-2Channel	1.0 210316	I/O-Controller
	NAIS-FP-Alarm-Circuit	1.03 171214	I/O-Controller
	NAIS-FP-Inv-Alarm-Circuit	1.01 171214	I/O-Controller
	NAIS-FP-Inv-Pulse-Output	1.01 171214	I/O-Controller
	NAIS-FP-Latching-Relay	1.03 171214	I/O-Controller
	NAIS-FP-Protection-Switch	1.04 171214	Redundancy-Controller
	NAIS-FP-Pulse-Output	1.01 171214	I/O-Controller
	NAIS-FP-Relay	1.02 171214	I/O-Controller
	NAIS-FP-RTD-Module	1.01 171214	I/O-Controller
	NAIS-FP-Simple-Input	1.02 171214	I/O-Controller

Vendor	Driver	Version	Tags
	NAIS-FP-Simple-Output	1.03 171214	I/O-Controller
	NAIS-FP-TxOn-Switch	1.01 171214	I/O-Controller
	NAIS-FP-Watchdog-Output	1.02 171214	I/O-Controller
	NAIS-FP0-REG	1.03 200113	I/O-Controller
	NAIS-FP0-RO	1.02 200113	I/O-Controller
	NAIS-FP0	1.02 171214	I/O-Controller
Narda	Narda-MCP	1.03 171220	Spectrum-Analyzer Measurement
	Narda-NRA	1.01 250910	Measurement Spectrum-Analyzer
	Narda-Upconverter	1.03 190725	Upconverter
NDSatCom	NDSatCom-1Axis-ACU	1.01 171214	Antenna-Controller
	NDSatCom-3Axis-ACU	1.02 171214	Antenna-Controller
	NDSatCom-ACU4100A-SNG	1.01 171214	Antenna-Controller
	NDSatCom-ACU4100B	1.02 171214	Antenna-Controller
	NDSatCom-ACU5000-SNG	3.11 240314	Antenna-Controller
	NDSatCom-ACU5030	1.05 180226	Antenna-Controller
	NDSatCom-ACU6030-REST	1.02 250212	Antenna-Controller

Vendor	Driver	Version	Tags
	NDSatCom-CPI	1.07 240314	HPA TWTA
	NDSatCom-KBand-Downconverter	2.07 200116	Downconverter
	NDSatCom-KBand-Upconverter	2.06 230715	Upconverter
	NDSatCom-KuBand-Downconverter	2.07 200116	Downconverter
	NDSatCom-KuBand-Upconverter	2.06 230715	Upconverter
	NDSatCom-KuHBand-Downconverter	2.07 200116	Downconverter
	NDSatCom-KuHBand-Upconverter	2.06 230715	Upconverter
	NDSatCom-KuLBand-Downconverter	2.07 200116	Downconverter
	NDSatCom-KuLBand-Upconverter	2.06 230715	Upconverter
	NDSatCom-RCU-SNMP	2.10 230626	Redundancy-Controller
	NDSatCom-SkyWAN-1070	1.12 190725	Modem
	NDSatCom-SkyWAN-IDU	2.10 190725	Modem
	NDSatCom-SkyWAN	1.01 171214	Modem
	NDSatCom-Step-Track	1.05 171214	Antenna-Controller
	NDSatCom-TWTA-BC30	1.02 240314	HPA TWTA

Vendor	Driver	Version	Tags
	NDSatCom-TWTA-REST	1.03 251001	HPA TWTA
	NDSatCom-TWTA-SNMP	2.13 240314	HPA TWTA
	NDSatCom-TWTA	2.12 241009	HPA TWTA
	ORB-Step-Track	1.01 171221	Antenna-Controller
Network	Network-Vikinx-VD3232-Matrix	1.00 191011	Switch-Matrix
Nevion	Nevion-Sublime	1.01 171004	Switch
	Nevion-TVG420	1.04 190313	IP-Gateway
	Nevion-TVG425	2.01 190219	IP-Gateway
Newtec	Newtec-AZ110-S2-Only	1.01 241008	Modulator
	Newtec-AZ110	1.08 251006	Modulator
	Newtec-AZ200-1-To-1	1.01 171220	Redundancy-Controller
	Newtec-AZ410	1.03 220513	Modem
	Newtec-AZ910	1.07 171214	Demodulator
	Newtec-EL470	1.04 190725	Modem
	Newtec-FRC0710	1.01 230601	Upconverter
	Newtec-FRC0750	1.05 240926	Block-Upconverter Combiner

Vendor	Driver	Version	Tags
	Newtec-M6100	1.21 251006	Modulator
	Newtec-MCX7000-ASIIN	1.01 210203	Gateway
	Newtec-MCX7000-ASIOUT	1.00 210203	Gateway
	Newtec-MCX7000-Base	1.00 191029	Gateway
	Newtec-MCX7000-BISS-Descr	1.01 210617	Gateway
	Newtec-MCX7000-Demodulator	1.06 250702	Demodulator
	Newtec-MCX7000-IPIN	1.03 210617	Gateway
	Newtec-MCX7000-IPOUT	1.03 210617	Gateway
	Newtec-MCX7000-Modulator	1.00 250704	Modulator
	Newtec-MCX7000-Protection	1.00 191028	Gateway
	Newtec-MDM6100	1.03 230105	Modulator
	Newtec-NTC2080	1.13 210916	Modulator
	Newtec-NTC2142	1.02 240314	Upconverter Downconverter
	Newtec-NTC2180	1.04 210916	Modulator
	Newtec-NTC2277	1.17 251001	Modulator
	Newtec-PEP	1.03 171214	Gateway

Vendor	Driver	Version	Tags
	Newtec-USS0202	1.02 211214	Redundancy-Controller
	Newtec-USS0212	1.00 211214	Redundancy-Controller
Novella	Novella-Blockdownconverter	1.00 210518	Block-Downconverter
	Novella-Blockupconverter	1.00 210518	Block-Upconverter
	Novella-RU03	1.00 240807	Redundancy-Controller
	Novella-RU04	1.03 240913	Redundancy-Controller
	Novella-Tracking-Receiver	1.03 180223	Tracking-Receiver
Novelsat	NovelSat-NS2000	1.07 240516	Demodulator
Novotronik	Novotronik-GTA5200	1.03 171004	Coupler
	Novotronik-GTS0118	1.01 171123	Switch-Controller
	Novotronik-GTS0752	1.01 171123	Switch-Controller
	Novotronik-GTS0752A	1.01 171123	Switch-Controller
	Novotronik-GTS4600	1.12 171123	Switch-Controller
	Novotronik-GTS5022	1.00 200121	Switch-Controller
	Novotronik-GTS5460	1.01 171121	Switch-Controller
	Novotronik-GTS5852	1.01 171123	Switch-Controller

Vendor	Driver	Version	Tags
	Novotronik-GTS6300	1.05 180119	LNA/LNB-Controller Redundancy-Controller
	Novotronik-GTS6303	1.03 220728	LNA/LNB-Controller Redundancy-Controller
	Novotronik-GTS6305	1.01 171123	LNA/LNB-Controller
	Novotronik-GTS6821	1.01 171123	Switch-Controller
	Novotronik-GTS6832	1.01 171123	Switch-Controller
	Novotronik-GTV2455	1.02 171123	Amplifier
	Novotronik-MAS3160	1.00 171228	Switch-Matrix
	Novotronik-MAS3240	1.00 200204	Switch-Matrix
	Novotronik-MAS3660	1.00 171228	Switch-Matrix
	Novotronik-MAS3670	1.00 200214	Switch-Matrix
	Novotronik-MAS4488	1.00 200617	Switch-Matrix
	Novotronik-MAS5321	1.03 221028	Switch-Matrix
	Novotronik-MAS5513	1.01 171123	Switch-Matrix
	Novotronik-MAS5514	1.00 171004	Switch-Matrix
	Novotronik-Matrix24	1.02 171123	Switch-Matrix
	Novotronik-Matrix32	1.00 171123	Switch-Matrix

Vendor	Driver	Version	Tags
	Novotronik-Matrix8	1.01 170926	Switch-Matrix
Novus Power Products	Novus-ND2310D	1.00 220225	Distribution-Amplifier
	Novus-NR2110-SNMP	1.01 231205	Frequency-Reference
	Novus-NR2110	1.02 170831	Frequency-Reference
NPC Systems	NPC-ACU-550	1.01 250825	Antenna-Controller
NTT Electronics	NTT-HVD6100	1.04 240314	IRD
	NTT-HVE9100	1.03 240314	Encoder
	NTT-HVE9210	1.14 240314	Encoder
OpenGear	OpenGear-Frame	1.03 231114	Main-Frame
Paradise Datacom	Paradise-RCP2-1100-SNMP	1.03 230526	Redundancy-Controller
	Paradise-RCP2-SNMP	1.02 171215	Redundancy-Controller
	Paradise-RCP2	1.02 211222	Redundancy-Controller
	Paradise-SSPA-2200AM	1.00 201005	HPA SSPA
	Paradise-SSPA-SNMP	2.07 190725	HPA SSPA
Peak Communications	Peak-Beacon-Receiver	1.03 231212	Tracking-Receiver
	Peak-Blockdownconverter	1.01 240314	Downconverter

Vendor	Driver	Version	Tags
	Peak-Blockupconverter	1.02 240314	Blockupconverter
	Peak-DBU200	1.00 190320	Redundancy-Controller Block-Downconverter
	Peak-DBU200H	1.01 190320	Redundancy-Controller Block-Downconverter
	Peak-Downconverter-7000	1.03 240314	Downconverter
	Peak-IBDH	1.01 240314	Redundancy-Controller
	Peak-P700	1.03 190725	Upconverter Downconverter
	Peak-UPC7002	1.09 240418	Uplink-Power-Control
	Peak-Upconverter-7000	1.03 240314	Upconverter
Pico Technology	Pico-EnviroMon-EL005	1.02 171215	Temperature-Sensor
Polsat	Polsat-ACTS	1.00 230228	Antenna-Controller
Profline	Profline-Plus1-AudioReceiver	1.02 200204	IRD Audio-Decoder
Promax	Promax-Sathunter	1.03 171215	Measurement
Qbit	Qbit-Q572-AudioReceiver	1.03 240314	IRD Audio-Decoder
Radeus Labs	RadeusLab-ACU-8200	1.02 220615	Antenna-Controller
Radyne Comstream	Radyne-DM240	1.05 240314	Modulator
	Radyne-DMD2050	1.04 190725	Modem

Vendor	Driver	Version	Tags
	Radyne-DVB3030	1.12 210916	Modulator
	Radyne-RCS11	1.02 171215	Redundancy-Controller
	Radyne-SFC1450	1.04 240314	Upconverter
Raritan	Raritan-EMX	1.00 190725	Temperature-Sensor Measurement
Research Concepts	ResearchConcepts-RC1500	1.03 190321	Antenna-Controller
	ResearchConcepts-RC2000	1.14 181004	Antenna-Controller
	ResearchConcepts-RC2500	1.01 180502	Antenna-Controller
	ResearchConcepts-RC3000	1.04 180322	Antenna-Controller
RF Design	RFDesign-Amp	1.00 180131	Amplifier
	RFDesign-FLCR-ODA	1.00 240912	Optical-Links
	RFDesign-FLCR	1.00 240912	Optical-Links
	RFDesign-FlexLink-S7B	1.00 220204	Switch-Matrix
	RFDesign-FlexLink-S7Bc	1.02 220826	Switch-Matrix
	RFDesign-LSC	1.04 240314	Splitter Combiner
	RFDesign-QLink	1.00 240912	Optical-Links
	RFDesign-RedLink	1.04 180322	Optical-Links

Vendor	Driver	Version	Tags
	RFDesign-RLC	1.00 190227	Combiner
	RFDesign-RLS	1.00 190227	Splitter
	RFDesign-RSC	1.02 171215	Redundancy-Controller
Rohde&Schwarz	RuS-PKU-SSPA	1.02 201026	SSPA
Romantis	Romantis-UHP	1.03 190725	Modem
RS	RuS-FPH	1.00 250910	Measurement Spectrum-Analyzer
	RuS-FSC	1.00 250910	Measurement Spectrum-Analyzer
	RuS-FSH	2.00 250910	Measurement Spectrum-Analyzer
	RuS-FSL	2.00 250910	Measurement Spectrum-Analyzer
	RuS-FSV	2.00 250910	Measurement Spectrum-Analyzer
S3 Satcom	S3Satcom-ACU	1.02 200518	Antenna-Controller
Satellink	Satellink-1to2-Redundant-LNA	1.02 171215	LNA/LNB-Controller Redundancy-Controller
Satellite Systems	SatelliteSystems-3430-BTR-L	1.01 240105	Tracking-Receiver
SatService	Antenna-BBU-Connector	1.00 230228	logical
	Antenna-Fixed	1.00 250401	logical
	Antenna-Management	1.09 251126	logical

Vendor	Driver	Version	Tags
	Antenna-Pointing-Fixed	1.03 171221	logical
	Antenna-Pointing-Hiltron	1.00 210124	logical
	Antenna-Pointing-ODM	2.00 181004	logical
	Antenna-Pointing	3.04 171221	logical
	Antenna-Proxy	1.01 171221	logical
	Antenna-Tracking	1.06 171221	logical
	Antenna-Wind-Protection	1.01 210427	logical
	APS-Backup-Antenna-Control	2.01 171221	logical
	APS-Backup-Antenna-Switch	2.01 171221	logical
	APS-Site-Switch	1.05 171221	logical
	ARD-Backup-Client	1.04 250318	logical
	ARD-Backup-Server	1.03 250318	logical
	Arithmetic	1.01 171221	logical
	Auto-RollOff	1.02 171221	logical
	WG-Switch	2.01 171221	logical
	AzElCollector	1.02 171221	logical

Vendor	Driver	Version	Tags
	Band-Pol-4-To-1	1.01 171221	logical
	BISS-KeySelect-CDB-DBonly	2.04 171221	logical
	BISS-KeySelect-CDB	2.04 171221	logical
	BISS-KeySelect	1.00 240213	logical
	Carrier-Site-Switch	1.07 221102	logical
	Channel-Select	1.01 171221	logical
	Choice-Translator	2.00 230116	logical
	Choice	1.05 171221	logical
	Collector-Two-Way	1.05 240528	logical
	Collector	1.06 171221	logical
	Comment	1.01 171221	logical
	Compare	1.03 171221	logical
	Conditional-Replicator	1.00 220201	logical
	RX-Channel	1.00 201021	logical
	De-Icing-Buttons	1.01 180608	logical
	De-Icing-Multiplex	2.01 180720	logical

Vendor	Driver	Version	Tags
	Delay-Line-Mirror	1.01 171221	logical
	Delay-Line	1.01 171221	logical
	Delayed-Gate-3	1.02 250207	logical
	Device-Copy	1.01 171221	logical
	Device-Mirror	1.01 171221	logical
	Device-Switcher	1.00 20220629	logical
	Downconverter-Protection-Controller	1.00 250318	logical
	Dual-Timezone-Clock	1.00 190528	logical
	Dual-Translator	1.01 171221	logical
	Dual-Wind-Speed	1.00 210427	logical
	ETL-MatrixPort-Emulator	1.01 250207	logical
	ETL-PDSwitch-Emulator	4.00 231211	logical
	Execute	2.0 211025	logical
	Fault-Array	1.01 171221	logical
	Fault-Gate-3	1.05 171221	logical
	Fault-Gate	1.04 171221	logical

Vendor	Driver	Version	Tags
	File-Recorder	2.03 190616	logical
	Filter	1.02 171221	logical
	Floating-Limit	1.01 200519	logical
	Frequency-Loop	1.04 171221	logical
	Frequency-Split	1.01 171221	logical
	Gate-Array-20	1.00 190528	logical
	Gate-Array	1.03 190506	logical
	Generic	1.01 171221	logical
	Hitec-NTP	1.0 201124	logical
	Inmarsat-NLES-Status	1.01 250917	logical
	Input	1.02 171221	logical
	IO-FEP2-Monitor	1.00 231213	logical
	IRD-Chain	1.02 171221	logical
	Level-Add	1.07 240822	logical
	Level-AddRed	1.07 240822	logical
	Level-Control	2.04 200310	logical

Vendor	Driver	Version	Tags
	Level-Monitoring	1.00 190523	logical
	Level-Set	2.08 200310	logical
	LineUp-Control	1.01 171221	logical
	Link-Mirror	1.02 171221	logical
	Link-Static	1.02 171221	logical
	Link-Two-Way	1.01 171221	logical
	Link	2.02 171221	logical
	Logger	1.01 171221	logical
	Logical-Gate-3	3.00 251029	logical
	Logical-Gate	3.00 251029	logical
	Macro-Player	1.03 171221	logical
	Matrix-Protection-Switcher	1.01 240313	logical
	Matrix-Site-Switch	2.01 190930	logical
	Measurement-Converter	1.01 171221	logical
	Modcod-Splitter	1.01 171221	logical
	Modem-Protection-Controller	1.05 250319	logical

Vendor	Driver	Version	Tags
	Monitor-Switch	1.02 171221	logical
	Multi-Add	1.00 250826	logical
	Multi-Protection-Switch	1.02 171221	logical
	Multi-Site-Switch	1.01 210211	logical
	Multi-Symmetric-Site-Switch	1.00 250805	logical
	N-Way-Switch-Latching	2.03 171221	logical
	N-Way-Switch	1.03 171221	logical
	Negate	1.02 171221	logical
	Noise-Source-Measurement	1.01 240104	logical
	Number-Select	1.01 171221	logical
	Number-Set	1.01 171221	logical
	Off-Switch	1.00 180115	logical
	OnTarget	1.01 230817	logical
	Output	1.05 180628	logical
	Parameter-Mirror	1.02 250909	logical
	Pass-Scheduler	1.05 230613	logical

Vendor	Driver	Version	Tags
	Pointing-Lock	1.03 171221	logical
	Pointing-Robot	2.05 180105	logical
	Polarization-Select	1.03 171221	logical
	PowerCalc-Transponder	1.02 241211	logical
	PowerCalc	1.02 171221	logical
	Protection-Switch-1-To-1-Dual	1.03 201119	logical
	Protection-Switch-1-To-1	2.05 221217	logical
	Protection-Switch-2-To-1	1.02 231027	logical
	Protection-Switch-Asymmetric	1.01 190705	logical
	Protection-Switch-N-To-1	4.07 241127	logical
	Protection-Switch-N-To-M	1.03 250318	logical
	Protection-Switch-NoSave-1-To-1	1.02 241217	logical
	Proxy-Device	1.01 171221	logical
	Remote-Alarm	1.02 171221	logical
	Replicator	3.00 221202	logical
	Report-Generator	2.00 190410	logical

Vendor	Driver	Version	Tags
	Reset-Timer	1.02 171221	logical
	RX-Channel-ActivationBased	2.01 251001	logical
	RX-Channel-ByChannel	2.03 251001	logical
	RX-Channel-BySatellite	2.03 251126	logical
	RX-Channel	1.15 230719	logical
	RX-ChannelSelect-CDB-DBonly	2.08 201019	logical
	RX-ChannelSelect-CDB	2.09 180129	logical
	RX-ChannelSelect	2.07 171221	logical
	RX-LineSettings-CDB-DBonly	1.06 171221	logical
	RX-LineSettings-CDB	1.05 171221	logical
	Sampler	1.01 171221	logical
	SatService-ACU-IDU	1.02 171215	Antenna-Controller
	SatService-ACU-ODM-Serial	1.06 230718	Antenna-Controller
	SatService-ACU-ODM	3.05 230719	Antenna-Controller
	SatService-ACU-Prediction	1.02 210621	logical
	SatService-ACU2-Serial	2.06 230920	Antenna-Controller

Vendor	Driver	Version	Tags
	SatService-ACU2	2.05 240423	Antenna-Controller
	SatService-Beacon-Receiver	4.02 230911	Tracking-Receiver
	SatService-Deice	1.04 191113	Deice-Controller
	SatService-IO-FEP-Protection-1-To-1	2.04 201119	Redundancy-Controller
	SatService-IO-FEP-Protection-2-To-1	1.04 241211	Redundancy-Controller
	SatService-IO-FEP	2.09 171026	I/O-Controller Redundancy-Controller Switch-Controller
	SatService-IO-FEP2	2.01 220823	I/O-Controller Redundancy-Controller Switch-Controller
	SatService-LBRX-1MT	1.02 171215	Tracking-Receiver
	SatService-LFRXTX	1.72 250725	Optical-Links
	SatService-Matrix	2.03 180424	Switch-Matrix
	SatService-Power-Sensor	1.01 171026	Powersensor Measurement
	SatService-SMU	1.01 161026	Main-Frame Distributor Redundancy-Controller Switch-Controller
	SatService-Temp-232	1.01 171026	Temperature-Sensor
	Selector-Two-Way	1.00 230210	logical
	Selector	1.03 201210	logical

Vendor	Driver	Version	Tags
	SES-Attn-Offset	1.01 171221	logical
	SES-DAKS	1.02 251028	logical
	SES-Level-GET	1.02 171221	logical
	SES-Level-SET	1.02 171221	logical
	SES-Power-Control-Box	1.02 250618	logical
	SES-Sysinfo	1.06 190521	logical
	Show-Orbit	1.01 171221	logical
	Site-Switch	1.02 171221	logical
	SNG-Pointing	1.02 171221	logical
	Spectrum-QAccess	1.04 200924	logical
	SRG-ChannelSelect	2.02 171221	logical
	Store-Float	1.00 231117	logical
	Stream-KeySelect	1.01 251001	logical
	Switch-Cross-Link	1.01 171221	logical
	Symmetric-Site-Switch	1.00 250730	logical
	Sync-Switch	1.02 171221	logical

Vendor	Driver	Version	Tags
	Sysinfo	3.15 250728	Server
	Temperature-Converter	1.03 220824	logical
	Threshold	1.01 171221	logical
	Time-Test	1.01 220920	logical
	TLE-Updater	1.00 210624	logical
	Toggle-Switch	1.01 171221	logical
	Transmit-Chain-Preset	2.01 171221	logical
	Triple-Translator	1.01 171221	logical
	TX-Channel-ByChannel	2.02 251001	logical
	TX-Channel-BySatellite	2.01 251001	logical
	TX-Channel	2.01 211014	logical
	TX-ChannelSelect-CDB	2.07 171221	logical
	TX-ChannelSelect	2.01 171221	logical
	TX-LineSettings-CDB	1.05 171221	logical
	TX-OFF-Timer	2.03 171221	logical
	UDP-Sender	1.00 190708	logical

Vendor	Driver	Version	Tags
	Uplink-Power-Control-2	1.02 171222	Uplink-Power-Control
	Uplink-Power-Control	1.12 241126	Uplink-Power-Control
	VioIP-End-Point	1.04 190128	logical
	Virtual-Matrix-128x128	1.01 171221	logical
	Virtual-Matrix-128x32	1.03 171221	logical
	WG-Switch	2.03 171221	logical
Scopus	Scopus-E1X00-SNMP	1.07 240314	Encoder Modulator
	Scopus-IRD26XX	1.03 171215	IRD
	Scopus-IRD28XX	1.07 171215	IRD
	Scopus-IRD2900-SNMP	3.01 171215	IRD
	Scopus-UE9000-SNMP	3.03 210916	Encoder Modulator
Sencore	Sencore-Afiniti-Encoder	1.01 251124	Encoder
	Sencore-AG4400	1.05 171215	IRD
	Sencore-MIP6210	1.03 240314	Recorder Player
	Sencore-MRD3187-DVBS2	1.04 171215	IRD
	Sencore-MRD3187-QUAD	2.01 230615	IRD

Vendor	Driver	Version	Tags
	Sencore-MRD5800	1.01 240314	IRD
	Sencore-MRD7000	1.14 250708	IRD
ServerTech	ServerTech-PDU-Sentry-3	1.01 180925	Power-Module
	ServerTech-PDU-Sentry-4	1.07 230912	Power-Module
Sicilsat	Sicilsat-S00761	1.00 250716	Antenna-Controller
Spacepath	Spacepath-CM2-TWTA	1.02 250305	HPA TWTA
	Spacepath-Tango-TWTA	1.00 240822	HPA TWTA
	Spacepath-TWTA-Serial	1.02 210125	HPA TWTA
	Spacepath-TWTA	1.07 240314	HPA TWTA
Speciality Microwave	Speciality-BDC-Controller	1.02 171220	Redundancy-Controller LNA/LNB-Controller
	Speciality-Controller-1926-100	1.02 240314	Redundancy-Controller
	Speciality-Controller-9753-500	1.01 240314	Redundancy-Controller
SPX	SPX-Etsiline-CommPact-v101	1.02 171010	Dehydrator
	SPX-Etsiline-CommPact	2.00 171024	Dehydrator
SSE	SSE-ASat	1.03 240314	Transceiver
	SSE-KStar	1.06 240314	Transceiver

Vendor	Driver	Version	Tags
Symmetricom	Symmetricom-4036B	1.02 190227	Distribution-Amplifier
	Symmetricom-4059B	1.00 190227	Distribution-Amplifier
	Symmetricom-9611B	1.02 171215	Distribution-Amplifier
	Symmetricom-S650	1.00 240418	GPS Frequency-Reference
	Symmetricom-XLI	1.01 240314	GPS Frequency-Reference
Synology	Synology-DSM	1.01 200624	Server
Tadiran	Tadiran-NTC2077	1.03 210916	Modulator
	Tadiran-Scopus-E1000	1.05 240314	Encoder
Tandberg	Tandberg-Alteia	1.03 171218	IRD
	Tandberg-E57XX-SNMP	4.03 210916	Encoder Modulator
	Tandberg-RX1290-SNMP	4.06 171218	IRD
	Tandberg-RX1290-SNMP-V3.9	3.01 171218	IRD
	Tandberg-RX8200-SNMP	1.34 251006	IRD
	Tandberg-RX8200-TriBand-SNMP	1.23 240314	IRD
	Tandberg-SM62xx	1.00 230209	Modulator
	Tandberg-TT1260-SNMP-PREV5	1.04 171218	IRD

Vendor	Driver	Version	Tags
	Tandberg-TT1260-SNMP	4.05 180206	IRD
	Tandberg-TT1260	1.02 171218	IRD
Technotrans	Technotrans-HC-Vertex	1.16 200730	Air-Conditioning
Teledyne Paradise	Teledyne-QFlex-400	1.01 250715	Modem
	Teledyne-QMultiFlex	1.01 250715	Modem
Teracue	Teracue-TAC	1.01 200912	Switch-Matrix
Terrasat	Terrasat-IBUC-SNMP	4.12 251124	Block-Upconverter
	Terrasat-IBUC	1.06 190725	Block-Upconverter
	Terrasat-RX-Protection	1.02 240314	Redundancy-Controller
TimeLink	TimeLink-Distributor-TMD	1.00 200515	Frequency-Reference Distributor
	TimeLink-Distributor	1.01 171218	Frequency-Reference Distributor
	TimeLink-GNSS-Receiver-TMG3400	1.01 200629	GPS Frequency-Reference
	TimeLink-GPS-Receiver	1.01 171218	GPS Frequency-Reference
	TimeLink-TMG3420	1.01 251113	GPS Frequency-Reference
	Timetech-GPS-Receiver	1.03 171218	GPS Frequency-Reference
Timetech	Timetech-Distributor	1.03 171218	Frequency-Reference Distributor

Vendor	Driver	Version	Tags
Tripp-Lite	TrippLite-PDU	1.02 240314	Power-Module
TurboSight	V4L-DVBS2-Demod	1.00 120116	Demodulator
TVIPS	Tvips-TNS541-SNMP	1.02 171218	Measurement
Vaisala	ColumbiaWeather-MicroServer	1.00 250714	Weather
	Vaisala-WXT	1.07 200916	Weather
Various	CSM-Spectrum-Analyzer	3.04 240104	Spectrum-Analyzer Measurement
Vertex	Vertex-7134-Step-Track	2.05 200219	Antenna-Controller
	Vertex-7200-Step-Track-DeltaPol	2.16 201208	Antenna-Controller
	Vertex-7200-Step-Track-Prec	2.16 201208	Antenna-Controller
	Vertex-7200-Step-Track	2.18 230817	Antenna-Controller
	Vertex-Digital-Tracking-Receiver	1.07 200412	Tracking-Receiver
	Vertex-Downconverter-legacy	1.06 251107	Downconverter
	Vertex-RSC-1100	1.01 171124	LNA/LNB-Controller Redundancy-Controller
	Vertex-RSC-1200	1.07 240314	LNA/LNB-Controller Redundancy-Controller
	Vertex-SCR-Downconverter-SNMP-legacy	1.03 251107	Downconverter

Vendor	Driver	Version	Tags
	Vertex-SCR-Downconverter	1.02 200116	Downconverter
	Vertex-SCR-Upconverter-SNMP-legacy	1.03 230323	Upconverter
	Vertex-SCR-Upconverter	1.02 190725	Upconverter
	Vertex-SSPA	2.05 240314	Block-Upconverter HPA SSPA
	Vertex-TWTA-2100TK	1.06 240314	TWTA HPA
	Vertex-Upconverter-legacy	1.04 230323	Upconverter
Vertex Antennentechnik	Vertex-ACU5000	1.12 250723	Antenna-Controller
	Vertex-ACU8100	1.09 180716	Antenna-Controller
ViaLite	ViaLite-HRA-Dual-EDFA	1.02 241126	Optical-Links Amplifier
	ViaLite-HRA-EDFA	1.05 250331	Optical-Links Amplifier
	ViaLite-HRS-Switch-4way	1.00 250403	Optical-Links Switch-Controller
	ViaLite-HRS-Switch	1.01 200110	Optical-Links Switch-Controller
	ViaLiteHD-FO	1.06 250618	Optical-Links
ViaSat	ViaSat-4100-ACU	1.09 240123	Antenna-Controller
	ViaSat-EBEM	1.04 250214	Modem
Viprinet	Viprinet-Hub-Redundany-HTTPS	1.00 171130	Router IP-Gateway

Vendor	Driver	Version	Tags
	Viprinet-HUB-SNMP	1.02 180119	Router IP-Gateway
	Viprinet-Rugged-Router-HTTPS	1.07 190605	Router IP-Gateway
	Viprinet-Rugged-Router-SNMP	1.04 190328	Router IP-Gateway
	Viprinet-Tunnel-SNMP	1.04 180119	Router IP-Gateway
Vitec	Vitec-MGW-Encoder	1.01 250213	Encoder Gateway
Vitec T21	T21-9261-Decoder	1.05 250630	IRD
	T21-9261-Encoder	1.01 250403	Encoder
Vocality	Vocality-FourWire	1.02 171220	Intercom
Walton	Walton-DDC-Deice	1.00 210825	Deice-Controller
	Walton-Deice	1.05 220923	Deice-Controller
Wavestream	Wavestream-BUC-SNMP	1.03 240314	Block-Upconverter
	Wavestream-BUC	1.02 190725	Block-Upconverter
Wegener	Wegener-UNITY44XX	1.04 171218	IRD
Wilmers	Wilmers-blueberryCOMPACT	1.05 180223	Weather
Wisi	WISI-Chameleon-DVBC	1.01 240314	IRD
Work Microwave	Work-Amplifier	1.02 171114	Amplifier

Vendor	Driver	Version	Tags
	Work-AT61-Modulator	1.03 240924	Modulator
	Work-DualSBD	1.09 201105	Block-Downconverter
	Work-DualSBU	1.11 240119	Block-Upconverter
	Work-DualSSBD	1.03 200116	Block-Downconverter
	Work-DualSSBU	1.04 190910	Block-Upconverter
	Work-DualVSCD	1.13 210721	Block-Downconverter
	Work-DualVSCU	1.12 210721	Block-Upconverter
	Work-MBU-MiniBuc	2.02 190725	Block-Upconverter
	Work-QuadSBD	1.04 200116	Block-Downconverter
	Work-QuadSBU	1.06 190910	Block-Upconverter
	Work-RSC	1.04 210630	Redundancy-Controller
	Work-RSCM-Dual	1.01 211222	Redundancy-Controller
	Work-RSCM-Quad	1.00 230607	Redundancy-Controller
	Work-RSCM-Triple	1.02 220417	Redundancy-Controller
	Work-RSCM	1.11 211222	Redundancy-Controller
	Work-SBD	1.03 200116	Block-Downconverter

Vendor	Driver	Version	Tags
	Work-SBU	1.05 190910	Block-Upconverter
	Work-SCD	1.04 200116	Downconverter
	Work-SCU	1.05 200219	Upconverter
	Work-SDM2-Modulator	1.08 190725	Modulator
	Work-SDM2-MT6-Modulator	1.03 190725	Modulator
	Work-SDM2-MT6-Modulator	1.03 190725	Modulator
	Work-SM2CU-Modulator	1.04 210916	Modulator
	Work-SSBD	2.02 200116	Block-Downconverter
	Work-SSBU	2.03 230901	Block-Upconverter
	Work-STLT2-KuK	1.02 181016	Testloop-Translator
	Work-TLT-LO	1.05 210721	Testloop-Translator
	Work-TLT	2.07 210721	Testloop-Translator
	Work-TripleVSCD	1.02 210721	Block-Downconverter
	Work-TriSBD	1.00 211112	Block-Downconverter
	Work-TriSBU	1.01 220228	Block-Upconverter
	Work-VCR	1.00 181015	Controller

Vendor	Driver	Version	Tags
	Work-VSCD	1.09 210721	Block-Downconverter
	Work-VSCSR	1.04 190725	Frequency-Reference
	Work-VSCU	1.13 221123	Block-Downconverter
	Work-VSM2CU-Modulator-Delay	1.15 221025	Modulator
	Work-VSM2CU-Modulator	1.13 220113	Modulator
	Work-VSSBD	2.05 210721	Block-Downconverter
	Work-VSSBU	2.04 210721	Block-Upconverter
Xicom	Xicom-KHPA	1.06 240314	HPA Klystron
	Xicom-TWTA-XT	2.04 240314	HPA TWTA
	Xicom-TWTA-XTD	1.12 240314	HPA TWTA
	Xicom-TWTA-XTDL	1.21 240314	HPA TWTA UPC
	Xicom-TWTA-XTRT	1.18 240314	HPA TWTA
	Xicom-TWTA	2.09 240314	TWTA HPA
XMW	XMW-Blockupconverter	1.00 240606	Blockupconverter
Xytronix	Xytronix-X600	1.03 230223	I/O-Controller Switch-Controller
Zodiac	Zodiac-Cortex-ACU	2.02 240419	Antenna-Controller

Vendor	Driver	Version	Tags
	Zodiac-Cortex-CRT-IFM	1.10 240527	Modulator Demodulator
	Zodiac-Cortex-CRT-IFR	1.08 240527	Modulator Demodulator
	Zodiac-Cortex-CRT-PRJ	1.05 230323	Modulator Demodulator
	Zodiac-Cortex-CRT-TCU	1.09 240313	Modulator Demodulator
	Zodiac-Cortex-CRT-TMS	1.05 240313	Modulator Demodulator
	Zodiac-Cortex-CRT-TMU	1.07 240821	Modulator Demodulator
	Zodiac-Cortex-CRT	1.06 240313	Modulator Demodulator
	Zodiac-Cortex-DTR-Cal	1.03 171220	Tracking-Receiver
	Zodiac-Cortex-DTR-Path	1.04 171220	Tracking-Receiver
	Zodiac-Cortex-DTR-Q	1.05 221220	Tracking-Receiver
	Zodiac-Cortex-DTR	1.09 200917	Tracking-Receiver
	Zodiac-Cortex-DTR	1.07 170914	Tracking-Receiver
	Zodiac-Cortex-DTR	1.08 190312	Tracking-Receiver

9.3 Device Drivers By Tag

Tag	Driver	Version	Vendor
Air-Conditioning	APKieffer-TWTA-AC	1.02 240314	AP-Kieffer
	ASCSignal-NGC-Hub-AC	1.00 190725	ASCSignal

Tag	*Driver*	*Version*	*Vendor*
	Bard-MC4002-AC	1.00 231130	Bard
	Dantherm-ComboCooling1000	1.01 250610	Dantherm
	Dantherm-TKS3000A	1.02 171213	Dantherm
	ECU-J0475	1.00 230810	ECU Corporation
	Emerson-AC-HPM	1.01 171213	Emerson Liebert
	Technotrans-HC-Vertex	1.16 200730	Technotrans
Amplifier	ETL-Alto-Amp	1.04 180814	ETL Systems
	Novotronik-GTV2455	1.02 171123	Novotronik
	RFDesign-Amp	1.00 180131	RF Design
	ViaLite-HRA-Dual-EDFA	1.02 241126	ViaLite
	ViaLite-HRA-EDFA	1.05 250331	ViaLite
	Work-Amplifier	1.02 171114	Work Microwave
Antenna-Controller	Andrew-APC100	1.04 171220	Andrew
	ASCSignal-NGC-ACU	1.07 190725	ASCSignal
	AvL-ITC-ACU	1.07 251104	AvL
	AvL-TracStar	1.08 180323	AvL Technologies

Tag	Driver	Version	Vendor
	Calian-MCI-ACU	1.05 250909	Calian
	CGC-ACU	1.18 240611	Comtech CGC
	ChinaSuncom-9293S3-ACU	1.00 231025	China Suncom
	GDSatcom-ACU930	1.01 250327	GD Satcom
	GDSatcom-ACU940	1.05 250521	GD Satcom
	GDSatcom-ACU950-MP	1.03 250407	GD Satcom
	GDSatcom-ACU950	1.14 250407	GD Satcom
	GDSatcom-ACU970	1.06 250407	GD Satcom
	GDSatcom-ACU980	1.01 250407	GD Satcom
	Hiltron-ACU2	1.07 250514	Hiltron
	Hitec-AFP-Deice	1.02 171213	Hitec
	Hitec-HACU-01-NAOS	1.22 230427	Hitec
	Hitec-HACU-01-V3.02	1.00 250723	Hitec
	Hitec-HACU-01-V3	2.01 250923	Hitec
	Hitec-HACU-01	1.09 211027	Hitec
	Monterey-140	1.01 171214	Chaparral Communications

Tag	Driver	Version	Vendor
	NDSatCom-1Axis-ACU	1.01 171214	NDSatCom
	NDSatCom-3Axis-ACU	1.02 171214	NDSatCom
	NDSatCom-ACU4100A-SNG	1.01 171214	NDSatCom
	NDSatCom-ACU4100B	1.02 171214	NDSatCom
	NDSatCom-ACU5000-SNG	3.11 240314	NDSatCom
	NDSatCom-ACU5030	1.05 180226	NDSatCom
	NDSatCom-ACU6030-REST	1.02 250212	NDSatCom
	NDSatCom-Step-Track	1.05 171214	NDSatCom
	NPC-ACU-550	1.01 250825	NPC Systems
	ORB-Step-Track	1.01 171221	NDSatCom
	Polsat-ACTS	1.00 230228	Polsat
	RadeusLab-ACU-8200	1.02 220615	Radeus Labs
	ResearchConcepts-RC1500	1.03 190321	Research Concepts
	ResearchConcepts-RC2000	1.14 181004	Research Concepts
	ResearchConcepts-RC2500	1.01 180502	Research Concepts
	ResearchConcepts-RC3000	1.04 180322	Research Concepts

Tag	*Driver*	*Version*	*Vendor*
	S3Satcom-ACU	1.02 200518	S3 Satcom
	SatService-ACU-IDU	1.02 171215	SatService
	SatService-ACU-ODM-Serial	1.06 230718	SatService
	SatService-ACU-ODM	3.05 230719	SatService
	SatService-ACU2-Serial	2.06 230920	SatService
	SatService-ACU2	2.05 240423	SatService
	Sicilsat-S00761	1.00 250716	Sicilsat
	Vertex-7134-Step-Track	2.05 200219	Vertex
	Vertex-7200-Step-Track-DeltaPol	2.16 201208	Vertex
	Vertex-7200-Step-Track-Prec	2.16 201208	Vertex
	Vertex-7200-Step-Track	2.18 230817	Vertex
	Vertex-ACU5000	1.12 250723	Vertex Antennentechnik
	Vertex-ACU8100	1.09 180716	Vertex Antennentechnik
	ViaSat-4100-ACU	1.09 240123	ViaSat
	Zodiac-Cortex-ACU	2.02 240419	Zodiac
Audio-Decoder	Profline-Plus1-AudioReceiver	1.02 200204	Profline

Tag	*Driver*	*Version*	*Vendor*
	Qbit-Q572-AudioReceiver	1.03 240314	Qbit
Block-Downconverter	ASCSignal-NGC-BDC	1.02 200309	ASCSignal
	Codan-Transceiver	1.04 240314	Codan
	Comtech-MBT5000	1.02 240314	Comtech
	Comtech-MBT5000A	1.00 250515	Comtech
	Novella-Blockdownconverter	1.00 210518	Novella
	Peak-DBU200	1.00 190320	Peak Communications
	Peak-DBU200H	1.01 190320	Peak Communications
	Work-DualSBD	1.09 201105	Work Microwave
	Work-DualSSBD	1.03 200116	Work Microwave
	Work-DualVSCD	1.13 210721	Work Microwave
	Work-QuadSBD	1.04 200116	Work Microwave
	Work-SBD	1.03 200116	Work Microwave
	Work-SSBD	2.02 200116	Work Microwave
	Work-TripleVSCD	1.02 210721	Work Microwave
	Work-TriSBD	1.00 211112	Work Microwave

Tag	*Driver*	*Version*	*Vendor*
	Work-VSCD	1.09 210721	Work Microwave
	Work-VSCU	1.13 221123	Work Microwave
	Work-VSSBD	2.05 210721	Work Microwave
Block-Upconverter	Acorde-BUC-ACCVT	1.06 240807	Acorde Technologies
	Agilis-BUC	1.01 190725	Agilis
	Codan-BUC-7586L	1.01 230323	Codan
	Codan-LBUC-6x00	2.04 240314	Codan
	Codan-Transceiver	1.04 240314	Codan
	Comtech-LPOD-SNMP	1.05 240314	Comtech
	Comtech-MBT5000	1.02 240314	Comtech
	Comtech-MBT5000A	1.00 250515	Comtech
	Irt-Pico-BUC-RCP	1.06 190725	IRT Technologies
	Irt-Pico-BUC-RCP	1.01 220816	IRT Technologies Spacepath
	MissionMicrowave-SSPA	0.06 240729	Mission Microwave
	Newtec-FRC0750	1.05 240926	Newtec
	Novella-Blockupconverter	1.00 210518	Novella

Tag	Driver	Version	Vendor
	Terrasat-IBUC-SNMP	4.12 251124	Terrasat
	Terrasat-IBUC	1.06 190725	Terrasat
	Vertex-SSPA	2.05 240314	Vertex
	Wavestream-BUC-SNMP	1.03 240314	Wavestream
	Wavestream-BUC	1.02 190725	Wavestream
	Work-DualSBU	1.11 240119	Work Microwave
	Work-DualSSBU	1.04 190910	Work Microwave
	Work-DualVSCU	1.12 210721	Work Microwave
	Work-MBU-MiniBuc	2.02 190725	Work Microwave
	Work-QuadSBU	1.06 190910	Work Microwave
	Work-SBU	1.05 190910	Work Microwave
	Work-SSBU	2.03 230901	Work Microwave
	Work-TriSBU	1.01 220228	Work Microwave
	Work-VSSBU	2.04 210721	Work Microwave
Blockupconverter	Peak-Blockupconverter	1.02 240314	Peak Communications
	XMW-Blockupconverter	1.00 240606	XMW

Tag	*Driver*	*Version*	*Vendor*
Camera	ACTi-ACM1231-Webcam	1.03 171212	ACTi
Combiner	ETL-Dextra-Combiner	1.01 240314	ETL Systems
	ETL-Genus-COM402	1.02 221108	ETL Systems
	Newtec-FRC0750	1.05 240926	Newtec
	RFDesign-LSC	1.04 240314	RF Design
	RFDesign-RLC	1.00 190227	RF Design
Controller	Work-VCR	1.00 181015	Work Microwave
Coupler	Novotronik-GTA5200	1.03 171004	Novotronik
Decoder	Comstream-DAC700-PRE	1.04 171212	Comstream
	Comstream-DAC700	1.03 171212	Comstream
	Haivision-Makito-X-Decoder	1.04 250910	Haivision
Dehydrator	Altec-Puregas-P210	1.00 240514	Altec Air
	Altec-Puregas-P550	1.01 240314	Altec
	Cibred-LAB4	1.01 190515	Cibred Sud
	ET-ADH-2A	1.02 210519	Environmental Technology
	ETI-ADH-NETCOM	1.07 231220	Environmental Technology

Tag	*Driver*	*Version*	*Vendor*
	ETI-ASM-1-Manifold	1.01 231103	Environmental Technology
	SPX-Etsiline-CommPact-v101	1.02 171010	SPX
	SPX-Etsiline-CommPact	2.00 171024	SPX
Deice-Controller	Antech-Deice	1.01 171220	Antech
	ASCSignal-NGC-Deice	1.00 190725	ASCSignal
	Hiltron-Deice	1.03 180115	Hiltron
	Hiltron-Multi-Deice	1.00 201023	Hiltron
	Hitec-AFP-Deice	1.02 171213	Hitec
	Hitec-HAFP-00-NAOS1	1.04 230418	Hitec
	Hitec-HAFP-00	1.04 190228	Hitec
	Hitec-HSCU-00	1.05 251126	Hitec
	SatService-Deice	1.04 191113	SatService
	Walton-DDC-Deice	1.00 210825	Walton
	Walton-Deice	1.05 220923	Walton
Demodulator	AntwerpSpace-G3-Demod	1.09 230310	Antwerpspace
	Comstream-Demodulator	1.01 171212	Comstream

Tag	Driver	Version	Vendor
	Comtech-CDM760	1.00 241210	Comtech
	Dektec-DTA2137	3.07 200114	Dektec
	Dektec-DTA2137	1.07 130408	Dektec
	Newtec-AZ910	1.07 171214	Newtec
	Newtec-MCX7000-Demodulator	1.06 250702	Newtec
	NovelSat-NS2000	1.07 240516	Novelsat
	V4L-DVBS2-Demod	1.00 120116	TurboSight
	Zodiac-Cortex-CRT-IFM	1.10 240527	Zodiac
	Zodiac-Cortex-CRT-IFR	1.08 240527	Zodiac
	Zodiac-Cortex-CRT-PRJ	1.05 230323	Zodiac
	Zodiac-Cortex-CRT-TCU	1.09 240313	Zodiac
	Zodiac-Cortex-CRT-TMS	1.05 240313	Zodiac
	Zodiac-Cortex-CRT-TMU	1.07 240821	Zodiac
	Zodiac-Cortex-CRT	1.06 240313	Zodiac
Distribution-Amplifier	DEV-4111-DA	1.01 231025	DEV Systemtechnik
	DEV-Distribution-2190	1.02 171006	DEV Systemtechnik

Tag	*Driver*	*Version*	*Vendor*
	ETL-Alto-DA	1.02 240314	ETL Systems
	ETL-Dextra-DistAmp	1.07 190717	ETL Systems
	ETL-Genus-DA-22512	1.07 240103	ETL Systems
	ETL-Stingray-FO-Base-Serial	1.00 201020	ETL Systems
	ETL-Stingray-FO-Base	1.00 200922	ETL Systems
	Meinberg-SDU	1.00 240411	Meinberg
	Novus-ND2310D	1.00 220225	Novus Power Products
	Symmetricom-4036B	1.02 190227	Symmetricom
	Symmetricom-4059B	1.00 190227	Symmetricom
	Symmetricom-9611B	1.02 171215	Symmetricom
Distributor	Endrun-FDC3300	1.02 170926	Endrun Technologies
	Lynx-DVD5420-ASI-Distr	1.02 171213	Lynx Technik
	Lynx-DVD5820-ASI-Distr	1.02 171213	Lynx Technik
	SatService-SMU	1.01 161026	SatService
	TimeLink-Distributor-TMD	1.00 200515	TimeLink
	TimeLink-Distributor	1.01 171218	TimeLink

Tag	*Driver*	*Version*	*Vendor*
	Timetech-Distributor	1.03 171218	Timetech
Downconverter	Comtech-DT4500-SNMP	1.02 200116	Comtech
	Comtech-DT4500	1.06 200116	Comtech
	Comtech-LBC4000	1.00 250428	Comtech
	Cross-UpDownconverter	1.03 190725	Cross Technology
	ETL-Genus-Downconverter	1.01 251020	ETL Systems
	Miteq-Downconverter-9900-SNMP	1.00 230126	Miteq
	Miteq-Downconverter-9900	1.00 211008	Miteq
	Miteq-LBand-Downconverter	1.03 200116	Miteq
	Miteq-Upconverter-9900-SNMP	1.00 230126	Miteq
	NDSatCom-KBand-Downconverter	2.07 200116	NDSatCom
	NDSatCom-KuBand-Downconverter	2.07 200116	NDSatCom
	NDSatCom-KuHBand-Downconverter	2.07 200116	NDSatCom
	NDSatCom-KuLBand-Downconverter	2.07 200116	NDSatCom
	Newtec-NTC2142	1.02 240314	Newtec
	Peak-Blockdownconverter	1.01 240314	Peak Communications

Tag	Driver	Version	Vendor
	Peak-Downconverter-7000	1.03 240314	Peak Communications
	Peak-P700	1.03 190725	Peak Communications
	Vertex-Downconverter-legacy	1.06 251107	Vertex
	Vertex-SCR-Downconverter-SNMP-legacy	1.03 251107	Vertex
	Vertex-SCR-Downconverter	1.02 200116	Vertex
	Work-SCD	1.04 200116	Work Microwave
Encoder	Adtec-Afinity-Encoder	1.03 230202	Adtec
	Adtec-EN81	1.11 210916	Adtec
	Ateme-Kyrion-CM5000	1.25 250526	Ateme
	Ateme-TitanEdge-Encoder	1.11 251203	Ateme
	Barco-RE4220	1.03 240314	Barco
	Comstream-DAC700-PRE	1.04 171212	Comstream
	Comstream-DAC700	1.03 171212	Comstream
	Envivio-4Caster-B3	1.04 171213	Envivo
	Ericsson-AVP	1.06 250224	Ericsson
	Ericsson-Voyager-II-v9.5	3.19 210916	Ericsson

Tag	Driver	Version	Vendor
	Ericsson-Voyager-II	4.03 210916	Ericsson
	Haivision-Makito-X	1.07 251112	Haivision
	Mediakind-CE1	1.01 250326	Mediakind
	NTT-HVE9100	1.03 240314	NTT Electronics
	NTT-HVE9210	1.14 240314	NTT Electronics
	Scopus-E1X00-SNMP	1.07 240314	Scopus
	Scopus-UE9000-SNMP	3.03 210916	Scopus
	Sencore-Afiniti-Encoder	1.01 251124	Sencore
	T21-9261-Encoder	1.01 250403	Vitec T21
	Tadiran-Scopus-E1000	1.05 240314	Tadiran
	Tandberg-E57XX-SNMP	4.03 210916	Tandberg
	Vitec-MGW-Encoder	1.01 250213	Vitec
Frequency-Reference	Cross-Frequency-Source	1.01 171213	Cross Technology
	ETL-Genus-100MHz	1.00 251020	ETL Systems
	ETL-Genus-10MHz	1.00 240315	ETL Systems
	FeiZyfer-GPS-Receiver	1.03 250703	FeiZyfer

Tag	Driver	Version	Vendor
	Meinberg-LantimeM300	1.02 221110	Meinberg
	Novus-NR2110-SNMP	1.01 231205	Novus Power Products
	Novus-NR2110	1.02 170831	Novus Power Products
	Symmetricom-S650	1.00 240418	Symmetricom
	Symmetricom-XLI	1.01 240314	Symmetricom
	TimeLink-Distributor-TMD	1.00 200515	TimeLink
	TimeLink-Distributor	1.01 171218	TimeLink
	TimeLink-GNSS-Receiver-TMG3400	1.01 200629	TimeLink
	TimeLink-GPS-Receiver	1.01 171218	TimeLink
	TimeLink-TMG3420	1.01 251113	TimeLink
	Timetech-Distributor	1.03 171218	Timetech
	Timetech-GPS-Receiver	1.03 171218	TimeLink
	Work-VSCSR	1.04 190725	Work Microwave
Gateway	Ateme-TitanEdge-Gateway	1.11 251203	Ateme
	Haivision-SRT-Gateway	1.07 251104	Haivision
	Newtec-MCX7000-ASIIN	1.01 210203	Newtec

Tag	*Driver*	*Version*	*Vendor*
	Newtec-MCX7000-ASIOUT	1.00 210203	Newtec
	Newtec-MCX7000-Base	1.00 191029	Newtec
	Newtec-MCX7000-BISS-Descr	1.01 210617	Newtec
	Newtec-MCX7000-IPIN	1.03 210617	Newtec
	Newtec-MCX7000-IPOUT	1.03 210617	Newtec
	Newtec-MCX7000-Protection	1.00 191028	Newtec
	Newtec-PEP	1.03 171214	Newtec
	Vitec-MGW-Encoder	1.01 250213	Vitec
Generator	BMS-IOM-Generator	1.01 1800810	BMS-IOM
GPS	FeiZyfer-GPS-Receiver	1.03 250703	FeiZyfer
	Meinberg-LantimeM300	1.02 221110	Meinberg
	NMEA-GPS-Receiver	1.01 171214	Generic
	Symmetricom-S650	1.00 240418	Symmetricom
	Symmetricom-XLI	1.01 240314	Symmetricom
	TimeLink-GNSS-Receiver-TMG3400	1.01 200629	TimeLink
	TimeLink-GPS-Receiver	1.01 171218	TimeLink

Tag	*Driver*	*Version*	*Vendor*
	TimeLink-TMG3420	1.01 251113	TimeLink
	Timetech-GPS-Receiver	1.03 171218	TimeLink
Graphics	Aten-VGA-Splitter	1.01 171212	Aten
	GundD-CompactCenter	1.01 171213	Guntermann und Drunck
HPA	Actia-Sodielec-DNT-HPA	1.08 240314	Actia-Sodielec
	Advantech-Genesis	1.01 241212	Advantech Wireless
	Advantech-SSPB-GCP	1.01 240314	Advantech Wireless
	Advantech-SSPB-MCP	1.01 240314	Advantech Wireless
	Advantech-SSPB-SNMP	1.06 250715	Advantech Wireless
	Advantech-SSPB	1.09 190725	Advantech Wireless
	CPI-350W-TWTA	2.05 240314	CPI
	CPI-CHPA	1.09 241009	CPI
	CPI-GaN-SSPA	1.01 240314	CPI
	CPI-KHPA-G3	2.11 240313	CPI
	CPI-KHPA-G4	2.15 250702	CPI
	CPI-LPA	2.06 240313	CPI

Tag	Driver	Version	Vendor
	CPI-Mini-ODU	1.27 241119	CPI
	CPI-SLPA	1.11 250610	CPI
	CPI-TouchPower	1.13 240917	CPI
	CPI-TWTA-Ka	2.07 240314	CPI
	CPI-TWTA	2.06 240314	CPI
	CPI-VZU-TWTA	1.04 240314	CPI
	E2V-TWTA	1.06 200728	E2V
	Hiltron-HCS-SSPA	1.03 190725	Hiltron
	Hiltron-HCS-TWTA	1.06 200728	Hiltron
	MCL-MT-TWTA-CSP	1.10 250107	MCL
	MCL-MT-TWTA	2.05 200728	MCL
	Mitec-TX	1.01 240314	Mitec
	NDSatCom-CPI	1.07 240314	NDSatCom
	NDSatCom-TWTA-BC30	1.02 240314	NDSatCom
	NDSatCom-TWTA-REST	1.03 251001	NDSatCom
	NDSatCom-TWTA-SNMP	2.13 240314	NDSatCom

Tag	Driver	Version	Vendor
	NDSatCom-TWTA	2.12 241009	NDSatCom
	Paradise-SSPA-2200AM	1.00 201005	Paradise Datacom
	Paradise-SSPA-SNMP	2.07 190725	Paradise Datacom
	Spacepath-CM2-TWTA	1.02 250305	Spacepath
	Spacepath-Tango-TWTA	1.00 240822	Spacepath
	Spacepath-TWTA-Serial	1.02 210125	Spacepath
	Spacepath-TWTA	1.07 240314	Spacepath
	Vertex-SSPA	2.05 240314	Vertex
	Vertex-TWTA-2100TK	1.06 240314	Vertex
	Xicom-KHPA	1.06 240314	Xicom
	Xicom-TWTA-XT	2.04 240314	Xicom
	Xicom-TWTA-XTD	1.12 240314	Xicom
	Xicom-TWTA-XTDL	1.21 240314	Xicom
	Xicom-TWTA-XTRT	1.18 240314	Xicom
	Xicom-TWTA	2.09 240314	Xicom
I/O-Controller	Actia-Sodielec-DSS1300	1.02 171212	Actia-Sodielec

Tag	Driver	Version	Vendor
	Advantech-ADAM5K	1.02 171212	Advantech
	Advantech-ADAM6K	1.01 171212	Advantech
	AutomationDirect-Terminator	2.00 190312	Automation-Direct
	BMS-IOM-Trend-DI	1.01 180810	BMS-IOM
	Brainbox-EDxxx	1.01 240314	Brainbox
	Fanuc-PLC9030	1.03 171213	Fanuc
	HWGroup-Poseidon2	1.01 230207	HW Group
	Moxa-ioLogik-1200	1.03 240314	Moxa
	Moxa-ioLogik-2214	1.00 240402	Moxa
	Moxa-ioLogik-4000	1.02 171124	Moxa
	Moxa-ioLogik-4200	1.03 171124	Moxa
	Moxa-ioLogik-Current	1.01 171124	Moxa
	Moxa-ioLogik-Temperature	1.01 171124	Moxa
	Moxa-ioThinx-4510	1.00 210608	Moxa
	NAIS-FP-8-Port-Switch	1.01 170926	NAIS/Panasonic
	NAIS-FP-AD-Test	1.01 171214	NAIS/Panasonic

Tag	*Driver*	*Version*	*Vendor*
	NAIS-FP-AD4-2Channel	1.0 210316	NAIS/Panasonic
	NAIS-FP-Alarm-Circuit	1.03 171214	NAIS/Panasonic
	NAIS-FP-Inv-Alarm-Circuit	1.01 171214	NAIS/Panasonic
	NAIS-FP-Inv-Pulse-Output	1.01 171214	NAIS/Panasonic
	NAIS-FP-Latching-Relay	1.03 171214	NAIS/Panasonic
	NAIS-FP-Pulse-Output	1.01 171214	NAIS/Panasonic
	NAIS-FP-Relay	1.02 171214	NAIS/Panasonic
	NAIS-FP-RTD-Module	1.01 171214	NAIS/Panasonic
	NAIS-FP-Simple-Input	1.02 171214	NAIS/Panasonic
	NAIS-FP-Simple-Output	1.03 171214	NAIS/Panasonic
	NAIS-FP-TxOn-Switch	1.01 171214	NAIS/Panasonic
	NAIS-FP-Watchdog-Output	1.02 171214	NAIS/Panasonic
	NAIS-FP0-REG	1.03 200113	NAIS/Panasonic
	NAIS-FP0-RO	1.02 200113	NAIS/Panasonic
	NAIS-FP0	1.02 171214	NAIS/Panasonic
	SatService-IO-FEP	2.09 171026	SatService

Tag	*Driver*	*Version*	*Vendor*
	SatService-IO-FEP2	2.01 220823	SatService
	Xytronix-X600	1.03 230223	Xytronix
Intercom	Vocality-FourWire	1.02 171220	Vocality
Interface-Monitor	MIB2-Ethernet-Switch	1.01 240523	Generic
	MIB2-Interface-Monitor	1.04 191010	Generic
	Netgear-M4300-Monitor	1.02 200206	Generic
IP-Gateway	Nevion-TVG420	1.04 190313	Nevion
	Nevion-TVG425	2.01 190219	Nevion
	Viprinet-Hub-Redundany-HTTPS	1.00 171130	Viprinet
	Viprinet-HUB-SNMP	1.02 180119	Viprinet
	Viprinet-Rugged-Router-HTTPS	1.07 190605	Viprinet
	Viprinet-Rugged-Router-SNMP	1.04 190328	Viprinet
	Viprinet-Tunnel-SNMP	1.04 180119	Viprinet
IRD	Adtec-Afiniti-Decoder	1.01 220423	Adtec
	Adtec-RD60	1.08 171212	Adtec
	Astro-EdgeStreamer	1.05 250708	ASTRO Strobel Kommunikationssysteme GmbH

Tag	Driver	Version	Vendor
	Ateme-DR5000	3.01 250606	Ateme
	Ateme-DR8400	1.10 240314	Ateme
	Ateme-TitanEdge-IRD	2.22 251126	Ateme
	Blankom-DRP393-SNMP	1.07 240314	Blankom
	Cisco-D9800-IRD	1.04 210608	Cisco Synamedia
	Comstream-ABR200	1.01 171212	Comstream
	Comstream-ABR202	1.05 171212	Comstream
	Envivio-4View-D1	1.02 171213	Envivo
	Evertz-7881-IRD	1.02 240314	Evertz
	Hilkom-BNova-SNMP	1.02 171213	Hilkom
	Mediakind-RX1	1.02 250526	Mediakind
	NTT-HVD6100	1.04 240314	NTT Electronics
	Profline-Plus1-AudioReceiver	1.02 200204	Profline
	Qbit-Q572-AudioReceiver	1.03 240314	Qbit
	Scopus-IRD26XX	1.03 171215	Scopus
	Scopus-IRD28XX	1.07 171215	Scopus

Tag	Driver	Version	Vendor
	Scopus-IRD2900-SNMP	3.01 171215	Scopus
	Sencore-AG4400	1.05 171215	Sencore
	Sencore-MRD3187-DVBS2	1.04 171215	Sencore
	Sencore-MRD3187-QUAD	2.01 230615	Sencore
	Sencore-MRD5800	1.01 240314	Sencore
	Sencore-MRD7000	1.14 250708	Sencore
	T21-9261-Decoder	1.05 250630	Vitec T21
	Tandberg-Alteia	1.03 171218	Tandberg
	Tandberg-RX1290-SNMP	4.06 171218	Tandberg
	Tandberg-RX1290-SNMP-V3.9	3.01 171218	Tandberg
	Tandberg-RX8200-SNMP	1.34 251006	Tandberg
	Tandberg-RX8200-TriBand-SNMP	1.23 240314	Tandberg
	Tandberg-TT1260-SNMP-PREV5	1.04 171218	Tandberg
	Tandberg-TT1260-SNMP	4.05 180206	Tandberg
	Tandberg-TT1260	1.02 171218	Tandberg
	Wegener-UNITY44XX	1.04 171218	Wegener

Tag	*Driver*	*Version*	*Vendor*
	WISI-Chameleon-DVBC	1.01 240314	Wisi
Klystron	CPI-KHPA-G3	2.11 240313	CPI
	CPI-KHPA-G4	2.15 250702	CPI
	Xicom-KHPA	1.06 240314	Xicom
LNA/LNB-Controller	Acorde-ACRS-IDU	1.00 250916	Acorde Technologies
	ASCSignal-NGC-RedundancyModule	2.01 240314	ASCSignal
	ETL-2773-XXYY	1.01 240314	ETL Systems
	ETL-Alto-Amp	1.04 180814	ETL Systems
	ETL-Genus-PRN	1.01 250925	ETL Systems
	ETL-Piranha	1.01 240314	ETL Systems
	GDSatcom-RSC1200	2.00 220317	GD Satcom
	Geosync-RCU	1.07 251030	Geosync Microwave
	Hiltron-HCS-Power	1.01 171213	Hiltron
	Maxtech-LRC-1200	1.04 220422	Maxtech
	Mitec-ARK-RX	1.00 190710	Mitec
	Novotronik-GTS6300	1.05 180119	Novotronik

Tag	*Driver*	*Version*	*Vendor*
	Novotronik-GTS6303	1.03 220728	Novotronik
	Novotronik-GTS6305	1.01 171123	Novotronik
	Satellink-1to2-Redundant-LNA	1.02 171215	Satellink
	Speciality-BDC-Controller	1.02 171220	Speciality Microwave
	Vertex-RSC-1100	1.01 171124	Vertex
	Vertex-RSC-1200	1.07 240314	Vertex
logical	Antenna-BBU-Connector	1.00 230228	SatService
	Antenna-Fixed	1.00 250401	SatService
	Antenna-Management	1.09 251126	SatService
	Antenna-Pointing-Fixed	1.03 171221	SatService
	Antenna-Pointing-Hiltron	1.00 210124	SatService
	Antenna-Pointing-ODM	2.00 181004	SatService
	Antenna-Pointing	3.04 171221	SatService
	Antenna-Proxy	1.01 171221	SatService
	Antenna-Tracking	1.06 171221	SatService
	Antenna-Wind-Protection	1.01 210427	SatService

Tag	Driver	Version	Vendor
	APS-Backup-Antenna-Control	2.01 171221	SatService
	APS-Backup-Antenna-Switch	2.01 171221	SatService
	APS-Site-Switch	1.05 171221	SatService
	ARD-Backup-Client	1.04 250318	SatService
	ARD-Backup-Server	1.03 250318	SatService
	Arithmetic	1.01 171221	SatService
	Auto-RollOff	1.02 171221	SatService
	WG-Switch	2.01 171221	SatService
	AzElCollector	1.02 171221	SatService
	Band-Pol-4-To-1	1.01 171221	SatService
	BISS-KeySelect-CDB-DBonly	2.04 171221	SatService
	BISS-KeySelect-CDB	2.04 171221	SatService
	BISS-KeySelect	1.00 240213	SatService
	Carrier-Site-Switch	1.07 221102	SatService
	Channel-Select	1.01 171221	SatService
	Choice-Translator	2.00 230116	SatService

Tag	Driver	Version	Vendor
	Choice	1.05 171221	SatService
	Collector-Two-Way	1.05 240528	SatService
	Collector	1.06 171221	SatService
	Comment	1.01 171221	SatService
	Compare	1.03 171221	SatService
	Conditional-Replicator	1.00 220201	SatService
	RX-Channel	1.00 201021	SatService
	De-Icing-Buttons	1.01 180608	SatService
	De-Icing-Multiplex	2.01 180720	SatService
	Delay-Line-Mirror	1.01 171221	SatService
	Delay-Line	1.01 171221	SatService
	Delayed-Gate-3	1.02 250207	SatService
	Device-Copy	1.01 171221	SatService
	Device-Mirror	1.01 171221	SatService
	Device-Switcher	1.00 20220629	SatService
	Downconverter-Protection-Controller	1.00 250318	SatService

Tag	*Driver*	*Version*	*Vendor*
	Dual-Timezone-Clock	1.00 190528	SatService
	Dual-Translator	1.01 171221	SatService
	Dual-Wind-Speed	1.00 210427	SatService
	ETL-MatrixPort-Emulator	1.01 250207	SatService
	ETL-PDSwitch-Emulator	4.00 231211	SatService
	Execute	2.0 211025	SatService
	Fault-Array	1.01 171221	SatService
	Fault-Gate-3	1.05 171221	SatService
	Fault-Gate	1.04 171221	SatService
	File-Recorder	2.03 190616	SatService
	Filter	1.02 171221	SatService
	Floating-Limit	1.01 200519	SatService
	Frequency-Loop	1.04 171221	SatService
	Frequency-Split	1.01 171221	SatService
	Gate-Array-20	1.00 190528	SatService
	Gate-Array	1.03 190506	SatService

Tag	*Driver*	*Version*	*Vendor*
	Generic	1.01 171221	SatService
	Hitec-NTP	1.0 201124	SatService
	Inmarsat-NLES-Status	1.01 250917	SatService
	Input	1.02 171221	SatService
	IO-FEP2-Monitor	1.00 231213	SatService
	IRD-Chain	1.02 171221	SatService
	Level-Add	1.07 240822	SatService
	Level-AddRed	1.07 240822	SatService
	Level-Control	2.04 200310	SatService
	Level-Monitoring	1.00 190523	SatService
	Level-Set	2.08 200310	SatService
	LineUp-Control	1.01 171221	SatService
	Link-Mirror	1.02 171221	SatService
	Link-Static	1.02 171221	SatService
	Link-Two-Way	1.01 171221	SatService
	Link	2.02 171221	SatService

Tag	Driver	Version	Vendor
	Logger	1.01 171221	SatService
	Logical-Gate-3	3.00 251029	SatService
	Logical-Gate	3.00 251029	SatService
	Macro-Player	1.03 171221	SatService
	Matrix-Protection-Switcher	1.01 240313	SatService
	Matrix-Site-Switch	2.01 190930	SatService
	Measurement-Converter	1.01 171221	SatService
	Modcod-Splitter	1.01 171221	SatService
	Modem-Protection-Controller	1.05 250319	SatService
	Monitor-Switch	1.02 171221	SatService
	Multi-Add	1.00 250826	SatService
	Multi-Protection-Switch	1.02 171221	SatService
	Multi-Site-Switch	1.01 210211	SatService
	Multi-Symmetric-Site-Switch	1.00 250805	SatService
	N-Way-Switch-Latching	2.03 171221	SatService
	N-Way-Switch	1.03 171221	SatService

Tag	*Driver*	*Version*	*Vendor*
	Negate	1.02 171221	SatService
	Noise-Source-Measurement	1.01 240104	SatService
	Number-Select	1.01 171221	SatService
	Number-Set	1.01 171221	SatService
	Off-Switch	1.00 180115	SatService
	OnTarget	1.01 230817	SatService
	Output	1.05 180628	SatService
	Parameter-Mirror	1.02 250909	SatService
	Pass-Scheduler	1.05 230613	SatService
	Pointing-Lock	1.03 171221	SatService
	Pointing-Robot	2.05 180105	SatService
	Polarization-Select	1.03 171221	SatService
	PowerCalc-Transponder	1.02 241211	SatService
	PowerCalc	1.02 171221	SatService
	Protection-Switch-1-To-1-Dual	1.03 201119	SatService
	Protection-Switch-1-To-1	2.05 221217	SatService

Tag	Driver	Version	Vendor
	Protection-Switch-2-To-1	1.02 231027	SatService
	Protection-Switch-Asymmetric	1.01 190705	SatService
	Protection-Switch-N-To-1	4.07 241127	SatService
	Protection-Switch-N-To-M	1.03 250318	SatService
	Protection-Switch-NoSave-1-To-1	1.02 241217	SatService
	Proxy-Device	1.01 171221	SatService
	Remote-Alarm	1.02 171221	SatService
	Replicator	3.00 221202	SatService
	Report-Generator	2.00 190410	SatService
	Reset-Timer	1.02 171221	SatService
	RX-Channel-ActivationBased	2.01 251001	SatService
	RX-Channel-ByChannel	2.03 251001	SatService
	RX-Channel-BySatellite	2.03 251126	SatService
	RX-Channel	1.15 230719	SatService
	RX-ChannelSelect-CDB-DBonly	2.08 201019	SatService
	RX-ChannelSelect-CDB	2.09 180129	SatService

Tag	*Driver*	*Version*	*Vendor*
	RX-ChannelSelect	2.07 171221	SatService
	RX-LineSettings-CDB-DBonly	1.06 171221	SatService
	RX-LineSettings-CDB	1.05 171221	SatService
	Sampler	1.01 171221	SatService
	SatService-ACU-Prediction	1.02 210621	SatService
	Selector-Two-Way	1.00 230210	SatService
	Selector	1.03 201210	SatService
	SES-Attn-Offset	1.01 171221	SatService
	SES-DAKS	1.02 251028	SatService
	SES-Level-GET	1.02 171221	SatService
	SES-Level-SET	1.02 171221	SatService
	SES-Power-Control-Box	1.02 250618	SatService
	SES-Sysinfo	1.06 190521	SatService
	Show-Orbit	1.01 171221	SatService
	Site-Switch	1.02 171221	SatService
	SNG-Pointing	1.02 171221	SatService

Tag	Driver	Version	Vendor
	Spectrum-QAccess	1.04 200924	SatService
	SRG-ChannelSelect	2.02 171221	SatService
	Store-Float	1.00 231117	SatService
	Stream-KeySelect	1.01 251001	SatService
	Switch-Cross-Link	1.01 171221	SatService
	Symmetric-Site-Switch	1.00 250730	SatService
	Sync-Switch	1.02 171221	SatService
	Temperature-Converter	1.03 220824	SatService
	Threshold	1.01 171221	SatService
	Time-Test	1.01 220920	SatService
	TLE-Updater	1.00 210624	SatService
	Toggle-Switch	1.01 171221	SatService
	Transmit-Chain-Preset	2.01 171221	SatService
	Triple-Translator	1.01 171221	SatService
	TX-Channel-ByChannel	2.02 251001	SatService
	TX-Channel-BySatellite	2.01 251001	SatService

Tag	*Driver*	*Version*	*Vendor*
	TX-Channel	2.01 211014	SatService
	TX-ChannelSelect-CDB	2.07 171221	SatService
	TX-ChannelSelect	2.01 171221	SatService
	TX-LineSettings-CDB	1.05 171221	SatService
	TX-OFF-Timer	2.03 171221	SatService
	UDP-Sender	1.00 190708	SatService
	VioIP-End-Point	1.04 190128	SatService
	Virtual-Matrix-128x128	1.01 171221	SatService
	Virtual-Matrix-128x32	1.03 171221	SatService
	WG-Switch	2.03 171221	SatService
Main-Frame	DEV-Frame	1.00 201019	DEV Systemtechnik
	ETL-Genus-Frame	1.05 250217	ETL Systems
	ETL-Modular-Frame-26128	1.04 180429	ETL Systems
	Evertz-7700-Frame	1.01 250901	Evertz
	Evertz-7800-Frame	1.02 240314	Evertz
	Hiltron-HCS-Frame	1.00 180110	Hiltron

Tag	*Driver*	*Version*	*Vendor*
	Lynx-DVD5820-Main	1.03 171213	Lynx Technik
	OpenGear-Frame	1.03 231114	OpenGear
	SatService-SMU	1.01 161026	SatService
Measurement	Agilent-Spectrum-Analyzer	1.04 171220	Agilent
	BMS-IOM-AHU	1.03 240314	BMS-IOM
	Calian-Decimator	1.06 240918	Calian
	CSM-Spectrum-Analyzer	3.04 240104	Various
	HP-Powermeter-Legacy	1.00 201021	Hewlett-Packard
	HP-Powermeter-SCPI	1.02 171220	Hewlett-Packard
	Janitza-UMG604	1.02 171220	Janitza
	KBR-Multimess-4F96	1.02 190212	KBR
	Keysight-FieldFox	1.1 250319	Keysight
	LPT-3000RX4	1.0 250326	LPT
	Narda-MCP	1.03 171220	Narda
	Narda-NRA	1.01 250910	Narda
	Promax-Sathunter	1.03 171215	Promax

Tag	*Driver*	*Version*	*Vendor*
	Raritan-EMX	1.00 190725	Raritan
	RuS-FPH	1.00 250910	RS
	RuS-FSC	1.00 250910	RS
	RuS-FSH	2.00 250910	RS
	RuS-FSL	2.00 250910	RS
	RuS-FSV	2.00 250910	RS
	SatService-Power-Sensor	1.01 171026	SatService
	Tvips-TNS541-SNMP	1.02 171218	TVIPS
Modem	AntwerpSpace-IBB-Modem	1.15 240314	Antwerpspace
	Comstream-CM601A-SCPC	1.06 190725	Comstream
	Comstream-CM701-MC	1.02 190725	Comstream
	Comstream-CM701	1.02 190725	Comstream
	Comstream-LoSpeed-Modem-SCPC	1.05 210916	Comstream
	Comstream-Modem-Card-SCPC	1.03 210916	Comstream
	Comtech-CDD564	1.00	Comtech
	Comtech-CDM570	1.06 200420	Comtech
	Comtech-CDM600	1.53 240314	Comtech

Tag	Driver	Version	Vendor
	EF-Data-SDM300	1.02 190725	EF-Data
	IDirect-5000	1.06 190725	iDirect
	NDSatCom-SkyWAN-1070	1.12 190725	NDSatCom
	NDSatCom-SkyWAN-IDU	2.10 190725	NDSatCom
	NDSatCom-SkyWAN	1.01 171214	NDSatCom
	Newtec-AZ410	1.03 220513	Newtec
	Newtec-EL470	1.04 190725	Newtec
	Radyne-DMD2050	1.04 190725	Radyne Comstream
	Romantis-UHP	1.03 190725	Romantis
	Teledyne-QFlex-400	1.01 250715	Teledyne Paradise
	Teledyne-QMultiFlex	1.01 250715	Teledyne Paradise
	ViaSat-EBEM	1.04 250214	ViaSat
Modulator	Adtec-EN81	1.11 210916	Adtec
	AntwerpSpace-G3-Mod	1.08 230310	Antwerpspace
	Ateme-Kyrion-CM5000	1.25 250526	Ateme
	Comstream-Modulator	1.03 210917	Comstream

Tag	Driver	Version	Vendor
	Comtech-CDM760	1.00 241210	Comtech
	Enensys-Vyper-Modulator	1.05 251104	Enensys
	Ericsson-AVP	1.06 250224	Ericsson
	Ericsson-Voyager-II-v9.5	3.19 210916	Ericsson
	Ericsson-Voyager-II	4.03 210916	Ericsson
	Miteq-DVM100	1.03 210916	Miteq
	Newtec-AZ110-S2-Only	1.01 241008	Newtec
	Newtec-AZ110	1.08 251006	Newtec
	Newtec-M6100	1.21 251006	Newtec
	Newtec-MCX7000-Modulator	1.00 250704	Newtec
	Newtec-MDM6100	1.03 230105	Newtec
	Newtec-NTC2080	1.13 210916	Newtec
	Newtec-NTC2180	1.04 210916	Newtec
	Newtec-NTC2277	1.17 251001	Newtec
	Radyne-DM240	1.05 240314	Radyne Comstream
	Radyne-DVB3030	1.12 210916	Radyne Comstream

Tag	Driver	Version	Vendor
	Scopus-E1X00-SNMP	1.07 240314	Scopus
	Scopus-UE9000-SNMP	3.03 210916	Scopus
	Tadiran-NTC2077	1.03 210916	Tadiran
	Tandberg-E57XX-SNMP	4.03 210916	Tandberg
	Tandberg-SM62xx	1.00 230209	Tandberg
	Work-AT61-Modulator	1.03 240924	Work Microwave
	Work-SDM2-Modulator	1.08 190725	Work Microwave
	Work-SDM2-MT6-Modulator	1.03 190725	Work Microwave
	Work-SDM2-MT6-Modulator	1.03 190725	Work Microwave
	Work-SM2CU-Modulator	1.04 210916	Work Microwave
	Work-VSM2CU-Modulator-Delay	1.15 221025	Work Microwave
	Work-VSM2CU-Modulator	1.13 220113	Work Microwave
	Zodiac-Cortex-CRT-IFM	1.10 240527	Zodiac
	Zodiac-Cortex-CRT-IFR	1.08 240527	Zodiac
	Zodiac-Cortex-CRT-PRJ	1.05 230323	Zodiac
	Zodiac-Cortex-CRT-TCU	1.09 240313	Zodiac

Tag	*Driver*	*Version*	*Vendor*
	Zodiac-Cortex-CRT-TMS	1.05 240313	Zodiac
	Zodiac-Cortex-CRT-TMU	1.07 240821	Zodiac
	Zodiac-Cortex-CRT	1.06 240313	Zodiac
Optical-Links	DEV-FO-RX	1.00 201019	DEV Systemtechnik
	DEV-FO-TX	1.00 201019	DEV Systemtechnik
	ETL-Genus-SRY-RX	1.00 240826	ETL Systems
	ETL-Genus-SRY-TX	1.01 250606	ETL Systems
	ETL-Stingray-FO-RX-Serial	1.00 201020	ETL Systems
	ETL-Stingray-FO-RX	1.01 211210	ETL Systems
	ETL-Stingray-FO-TX-Serial	1.00 201020	ETL Systems
	ETL-Stingray-FO-TX	1.01 211210	ETL Systems
	Evertz-7703-BPX	1.02 190716	Evertz
	Evertz-7707-ET	1.00 181206	Evertz
	Evertz-7707-LR	1.01 190321	Evertz
	Evertz-7707-LT	1.01 190321	Evertz
	Evertz-7708-LR	1.01 190321	Evertz

Tag	*Driver*	*Version*	*Vendor*
	Evertz-7708-LT	1.00 190321	Evertz
	Foxcom-SatLight-Platinum	1.12 190515	Foxcom
	Hiltron-HCS-Power	1.01 171213	Hiltron
	RFDesign-FLCR-ODA	1.00 240912	RF Design
	RFDesign-FLCR	1.00 240912	RF Design
	RFDesign-QLink	1.00 240912	RF Design
	RFDesign-RedLink	1.04 180322	RF Design
	SatService-LFRXTX	1.72 250725	SatService
	ViaLite-HRA-Dual-EDFA	1.02 241126	ViaLite
	ViaLite-HRA-EDFA	1.05 250331	ViaLite
	ViaLite-HRS-Switch-4way	1.00 250403	ViaLite
	ViaLite-HRS-Switch	1.01 200110	ViaLite
	ViaLiteHD-FO	1.06 250618	ViaLite
Player	Sencore-MIP6210	1.03 240314	Sencore
Power-Module	APC-ATS	1.02 240314	APC
	APC-RPDU	1.03 190130	APC

Tag	Driver	Version	Vendor
	Gude-ExpertPowerControl	1.01 171213	Gude
	Infratec-PM	1.03 171213	Infratec
	ServerTech-PDU-Sentry-3	1.01 180925	ServerTech
	ServerTech-PDU-Sentry-4	1.07 230912	ServerTech
	TrippLite-PDU	1.02 240314	Tripp-Lite
Power-Supply	Geosync-PS	1.00 181105	Geosync Microwave
	Hiltron-HCS-PS	1.01 180117	Hiltron
Powersensor	BMS-IOM-Powermeter	1.01 180810	BMS-IOM
	BMS-IOM-Socomec-Powermeter	1.01 180810	BMS-IOM
	HP-Powermeter-Legacy	1.00 201021	Hewlett-Packard
	HP-Powermeter-SCPI	1.02 171220	Hewlett-Packard
	Janitza-UMG604	1.02 171220	Janitza
	KBR-Multimess-4F96	1.02 190212	KBR
	SatService-Power-Sensor	1.01 171026	SatService
Recorder	Sencore-MIP6210	1.03 240314	Sencore
Redundancy-Controller	Acorde-ACRS-IDU	1.00 250916	Acorde Technologies

Tag	Driver	Version	Vendor
	Anacom-Protection-Switch	1.01 171212	Anacom
	Apollo-Switch	1.02 171212	Apollo
	ASCSignal-NGC-RedundancyModule	2.01 240314	ASCSignal
	CandM-Switch-30510	1.00 220928	CandM
	CandM-Switch-308901	1.00 220621	CandM
	CandM-Switch	1.02 210819	CandM
	Codan-7586L	1.010 230323	Codan CPI
	Comstream-CX101	0.01 171212	Comstream
	Comstream-CX801	1.02 171212	Comstream
	CPI-SRCU-BCIP	1.01 240314	CPI
	CPI-SRCU	1.00 190326	CPI
	CPI-Switch	1.03 210819	CPI
	DoubleD-DDA89	1.00 251114	DoubleD Electronics
	GDSatcom-RSC1200	2.00 220317	GD Satcom
	Geosync-RCU-R2R	1.01 210924	Geosync Microwave
	Geosync-RCU	1.07 251030	Geosync Microwave

Tag	Driver	Version	Vendor
	Hiltron-HCS-Switch	1.01 171213	Hiltron
	Maxtech-LRC-1200	1.04 220422	Maxtech
	Mitec-ARK-RX	1.00 190710	Mitec
	Miteq-NSU	1.04 250718	Miteq
	NAIS-FP-Protection-Switch	1.04 171214	NAIS/Panasonic
	NDSatCom-RCU-SNMP	2.10 230626	NDSatCom
	Newtec-AZ200-1-To-1	1.01 171220	Newtec
	Newtec-USS0202	1.02 211214	Newtec
	Newtec-USS0212	1.00 211214	Newtec
	Novella-RU03	1.00 240807	Novella
	Novella-RU04	1.03 240913	Novella
	Novotronik-GTS6300	1.05 180119	Novotronik
	Novotronik-GTS6303	1.03 220728	Novotronik
	Paradise-RCP2-1100-SNMP	1.03 230526	Paradise Datacom
	Paradise-RCP2-SNMP	1.02 171215	Paradise Datacom
	Paradise-RCP2	1.02 211222	Paradise Datacom

Tag	Driver	Version	Vendor
	Peak-DBU200	1.00 190320	Peak Communications
	Peak-DBU200H	1.01 190320	Peak Communications
	Peak-IBDH	1.01 240314	Peak Communications
	Radyne-RCS11	1.02 171215	Radyne Comstream
	RFDesign-RSC	1.02 171215	RF Design
	Satellink-1to2-Redundant-LNA	1.02 171215	Satellink
	SatService-IO-FEP-Protection-1-To-1	2.04 201119	SatService
	SatService-IO-FEP-Protection-2-To-1	1.04 241211	SatService
	SatService-IO-FEP	2.09 171026	SatService
	SatService-IO-FEP2	2.01 220823	SatService
	SatService-SMU	1.01 161026	SatService
	Speciality-BDC-Controller	1.02 171220	Speciality Microwave
	Speciality-Controller-1926-100	1.02 240314	Speciality Microwave
	Speciality-Controller-9753-500	1.01 240314	Speciality Microwave
	Terrasat-RX-Protection	1.02 240314	Terrasat
	Vertex-RSC-1100	1.01 171124	Vertex

Tag	*Driver*	*Version*	*Vendor*
	Vertex-RSC-1200	1.07 240314	Vertex
	Work-RSC	1.04 210630	Work Microwave
	Work-RSCM-Dual	1.01 211222	Work Microwave
	Work-RSCM-Quad	1.00 230607	Work Microwave
	Work-RSCM-Triple	1.02 220417	Work Microwave
	Work-RSCM	1.11 211222	Work Microwave
Router	Cisco-2800GSM-SNMP	1.03 171212	Cisco
	Cisco-2900-SNMP	1.04 171212	Cisco
	Cisco-Monitor	1.02 171212	Cisco
	Cisco-Router	1.01 171212	Cisco
	Viprinet-Hub-Redundany-HTTPS	1.00 171130	Viprinet
	Viprinet-HUB-SNMP	1.02 180119	Viprinet
	Viprinet-Rugged-Router-HTTPS	1.07 190605	Viprinet
	Viprinet-Rugged-Router-SNMP	1.04 190328	Viprinet
	Viprinet-Tunnel-SNMP	1.04 180119	Viprinet
Server	Synology-DSM	1.01 200624	Synology

Tag	*Driver*	*Version*	*Vendor*
	Sysinfo	3.15 250728	SatService
Spectrum-Analyzer	Agilent-Spectrum-Analyzer	1.04 171220	Agilent
	Calian-Decimator	1.06 240918	Calian
	CSM-Spectrum-Analyzer	3.04 240104	Various
	Keysight-FieldFox	1.1 250319	Keysight
	LPT-3000RX4	1.0 250326	LPT
	Narda-MCP	1.03 171220	Narda
	Narda-NRA	1.01 250910	Narda
	RuS-FPH	1.00 250910	RS
	RuS-FSC	1.00 250910	RS
	RuS-FSH	2.00 250910	RS
	RuS-FSL	2.00 250910	RS
	RuS-FSV	2.00 250910	RS
Splitter	ETL-Genus-ALT	1.00 230526	ETL Systems
	ETL-Genus-DIV401	1.02 210521	ETL Systems
	ETL-Splitter-DIV409	1.03 171220	ETL Systems

Tag	*Driver*	*Version*	*Vendor*
	ETL-Splitter-DIV429	1.00 190920	ETL Systems
	RFDesign-LSC	1.04 240314	RF Design
	RFDesign-RLS	1.00 190227	RF Design
SSPA	Advantech-Genesis	1.01 241212	Advantech Wireless
	Advantech-SSPB-GCP	1.01 240314	Advantech Wireless
	Advantech-SSPB-MCP	1.01 240314	Advantech Wireless
	Advantech-SSPB-SNMP	1.06 250715	Advantech Wireless
	Advantech-SSPB	1.09 190725	Advantech Wireless
	CPI-GaN-SSPA	1.01 240314	CPI
	GDSatcom-Rack-SSPA	1.00 240314	GDSatcom CPI
	GDSatcom-SSPA-QuadMod	1.02 240314	GDSatcom
	Hiltron-HCS-SSPA	1.03 190725	Hiltron
	MissionMicrowave-SSPA	0.06 240729	Mission Microwave
	Mitec-TX	1.01 240314	Mitec
	Paradise-SSPA-2200AM	1.00 201005	Paradise Datacom
	Paradise-SSPA-SNMP	2.07 190725	Paradise Datacom

Tag	*Driver*	*Version*	*Vendor*
	RuS-PKU-SSPA	1.02 201026	Rohde&Schwarz
	Vertex-SSPA	2.05 240314	Vertex
Storage-Monitor	MIB2-Storage-Monitor	1.01 250224	Generic
Switch	DEV-Switch-2-To-1	1.00 201202	DEV Systemtechnik
	Nevion-Sublime	1.01 171004	Nevion
Switch-Controller	ETL-LS-Switch	1.09 241213	ETL Systems
	ETL-POE-Switch	1.01 240314	ETL Systems
	ETL-SHF-Switch-2to2	1.00 250428	ETL Systems
	ETL-SHF-Switch	1.02 171220	ETL Systems
	ETL-SwitchUnit-23258	1.03 180814	ETL Systems
	MiniCircuits-Switch-ZT	1.02 240314	MiniCircuits
	Mirad-VPC	1.03 171213	Mirad Microwave
	Mitec-Switch	1.02 171213	Mitec
	Novotronik-GTS0118	1.01 171123	Novotronik
	Novotronik-GTS0752	1.01 171123	Novotronik
	Novotronik-GTS0752A	1.01 171123	Novotronik

Tag	*Driver*	*Version*	*Vendor*
	Novotronik-GTS4600	1.12 171123	Novotronik
	Novotronik-GTS5022	1.00 200121	Novotronik
	Novotronik-GTS5460	1.01 171121	Novotronik
	Novotronik-GTS5852	1.01 171123	Novotronik
	Novotronik-GTS6821	1.01 171123	Novotronik
	Novotronik-GTS6832	1.01 171123	Novotronik
	SatService-IO-FEP	2.09 171026	SatService
	SatService-IO-FEP2	2.01 220823	SatService
	SatService-SMU	1.01 161026	SatService
	ViaLite-HRS-Switch-4way	1.00 250403	ViaLite
	ViaLite-HRS-Switch	1.01 200110	ViaLite
	Xytronix-X600	1.03 230223	Xytronix
Switch-Matrix	DEV-Matrix-1986	1.06 240314	DEV Systemtechnik
	DEV-Matrix-1996	1.00 250827	DEV Systemtechnik
	ETL-Genus-VTR-101-Matrix	1.02 240428	ETL Systems
	ETL-Genus-VTRC-101-Matrix	1.02 240428	ETL Systems

Tag	Driver	Version	Vendor
	ETL-Hurricane-Matrix	1.01 190403	ETL Systems
	ETL-NGM100-Matrix	2.07 250527	ETL Systems
	ETL-NGM100C-Matrix	1.02 250519	ETL Systems
	ETL-NGM21C-Matrix	1.12 250519	ETL Systems
	ETL-NGM30-Matrix	1.13 250519	ETL Systems
	ETL-NSN100-Matrix	4.02 250527	ETL Systems
	ETL-Volcano-Matrix	1.03 240314	ETL Systems
	ETL-VTR70-Matrix	1.04 251006	ETL Systems
	ETL-VTR70C-Matrix	1.03 251021	ETL Systems
	ETL-VTX100-Matrix	1.03 240702	ETL Systems
	Evertz-EMR-Matrix	1.00 250204	Evertz
	Harris-Panacea	1.02 171213	Harris
	Leitch-Panacea	1.01 171213	Leitch
	Network-Vikinx-VD3232-Matrix	1.00 191011	Network
	Novotronik-MAS3160	1.00 171228	Novotronik
	Novotronik-MAS3240	1.00 200204	Novotronik

Tag	*Driver*	*Version*	*Vendor*
	Novotronik-MAS3660	1.00 171228	Novotronik
	Novotronik-MAS3670	1.00 200214	Novotronik
	Novotronik-MAS4488	1.00 200617	Novotronik
	Novotronik-MAS5321	1.03 221028	Novotronik
	Novotronik-MAS5513	1.01 171123	Novotronik
	Novotronik-MAS5514	1.00 171004	Novotronik
	Novotronik-Matrix24	1.02 171123	Novotronik
	Novotronik-Matrix32	1.00 171123	Novotronik
	Novotronik-Matrix8	1.01 170926	Novotronik
	RFDesign-FlexLink-S7B	1.00 220204	RF Design
	RFDesign-FlexLink-S7Bc	1.02 220826	RF Design
	SatService-Matrix	2.03 180424	SatService
	Teracue-TAC	1.01 200912	Teracue
Synthesizer	HP-Synthesizer-SCPI	1.03 220823	Hewlett-Packard
Temperature-Sensor	ASCSignal-NGC-Temperature	1.00 190725	ASCSignal
	ComWinTop-THxxS	1.02 250910	ComWinTop

Tag	*Driver*	*Version*	*Vendor*
	HWGroup-Poseidon2	1.01 230207	HW Group
	HWGroup-STE	1.01 230209	HW Group
	HWGroup-STE2	1.03 180223	HW Group
	IP-Sens-IPS2225P	1.00 240411	IP Sens
	Moxa-ioLogik-Temperature	1.01 171124	Moxa
	Pico-EnviroMon-EL005	1.02 171215	Pico Technology
	Raritan-EMX	1.00 190725	Raritan
	SatService-Temp-232	1.01 171026	SatService
Testloop-Translator	AtlanTecRF-TLT	1.02 220929	Atlantic-Microwave
	Geosync-TLT	1.04 190725	Geosync Microwave
	Miteq-TLT-LO	1.00 210517	Miteq
	Miteq-TLT	1.01 210928	Miteq
	Work-STLT2-KuK	1.02 181016	Work Microwave
	Work-TLT-LO	1.05 210721	Work Microwave
	Work-TLT	2.07 210721	Work Microwave
Tracking-Receiver	AtlanticSat-ASC300	1.02 230911	Atlantic Satellite Corporation

Tag	Driver	Version	Vendor
	ChinaSuncom-Beacon-Receiver-SC9800	1.00 231030	China Suncom
	GDSatcom-DTR520	1.03 240916	GD Satcom
	Novella-Tracking-Receiver	1.03 180223	Novella
	Peak-Beacon-Receiver	1.03 231212	Peak Communications
	SatelliteSystems-3430-BTR-L	1.01 240105	Satellite Systems
	SatService-Beacon-Receiver	4.02 230911	SatService
	SatService-LBRX-1MT	1.02 171215	SatService
	Vertex-Digital-Tracking-Receiver	1.07 200412	Vertex
	Zodiac-Cortex-DTR-Cal	1.03 171220	Zodiac
	Zodiac-Cortex-DTR-Path	1.04 171220	Zodiac
	Zodiac-Cortex-DTR-Q	1.05 221220	Zodiac
	Zodiac-Cortex-DTR	1.09 200917	Zodiac
	Zodiac-Cortex-DTR	1.07 170914	Zodiac
	Zodiac-Cortex-DTR	1.08 190312	Zodiac
Transceiver	Anacom-KuBand-Transceiver	1.05 240314	Anacom
	SSE-ASat	1.03 240314	SSE

Tag	*Driver*	*Version*	*Vendor*
	SSE-KStar	1.06 240314	SSE
TWTA	Actia-Sodielec-DNT-HPA	1.08 240314	Actia-Sodielec
	CPI-350W-TWTA	2.05 240314	CPI
	CPI-CHPA	1.09 241009	CPI
	CPI-LPA	2.06 240313	CPI
	CPI-Mini-ODU	1.27 241119	CPI
	CPI-SLPA	1.11 250610	CPI
	CPI-TouchPower	1.13 240917	CPI
	CPI-TWTA-Ka	2.07 240314	CPI
	CPI-TWTA	2.06 240314	CPI
	CPI-VZU-TWTA	1.04 240314	CPI
	E2V-TWTA	1.06 200728	E2V
	Hiltron-HCS-TWTA	1.06 200728	Hiltron
	MCL-MT-TWTA-CSP	1.10 250107	MCL
	MCL-MT-TWTA	2.05 200728	MCL
	NDSatCom-CPI	1.07 240314	NDSatCom

Tag	*Driver*	*Version*	*Vendor*
	NDSatCom-TWTA-BC30	1.02 240314	NDSatCom
	NDSatCom-TWTA-REST	1.03 251001	NDSatCom
	NDSatCom-TWTA-SNMP	2.13 240314	NDSatCom
	NDSatCom-TWTA	2.12 241009	NDSatCom
	Spacepath-CM2-TWTA	1.02 250305	Spacepath
	Spacepath-Tango-TWTA	1.00 240822	Spacepath
	Spacepath-TWTA-Serial	1.02 210125	Spacepath
	Spacepath-TWTA	1.07 240314	Spacepath
	Vertex-TWTA-2100TK	1.06 240314	Vertex
	Xicom-TWTA-XT	2.04 240314	Xicom
	Xicom-TWTA-XTD	1.12 240314	Xicom
	Xicom-TWTA-XTDL	1.21 240314	Xicom
	Xicom-TWTA-XTRT	1.18 240314	Xicom
	Xicom-TWTA	2.09 240314	Xicom
UPC	Xicom-TWTA-XTDL	1.21 240314	Xicom
Upconverter	Comtech-LBC4000	1.00 250428	Comtech

Tag	Driver	Version	Vendor
	Comtech-UT4500-SNMP	1.02 190725	Comtech
	Comtech-UT4500	1.07 250724	Comtech
	Cross-UpDownconverter	1.03 190725	Cross Technology
	ETL-Genus-Upconverter	1.01 251020	ETL Systems
	Miteq-KBand-Upconverter	1.02 190725	Miteq
	Miteq-KuBand-Upconverter	1.00 210512	Miteq
	Miteq-LBand-Upconverter	1.03 190725	Miteq
	Miteq-NSU	1.04 250718	Miteq
	Narda-Upconverter	1.03 190725	Narda
	NDSatCom-KBand-Upconverter	2.06 230715	NDSatCom
	NDSatCom-KuBand-Upconverter	2.06 230715	NDSatCom
	NDSatCom-KuHBand-Upconverter	2.06 230715	NDSatCom
	NDSatCom-KuLBand-Upconverter	2.06 230715	NDSatCom
	Newtec-FRC0710	1.01 230601	Newtec
	Newtec-NTC2142	1.02 240314	Newtec
	Peak-P700	1.03 190725	Peak Communications

Tag	*Driver*	*Version*	*Vendor*
	Peak-Upconverter-7000	1.03 240314	Peak Communications
	Radyne-SFC1450	1.04 240314	Radyne Comstream
	Vertex-SCR-Upconverter-SNMP-legacy	1.03 230323	Vertex
	Vertex-SCR-Upconverter	1.02 190725	Vertex
	Vertex-Upconverter-legacy	1.04 230323	Vertex
	Work-SCU	1.05 200219	Work Microwave
Uplink-Power-Control	Miteq-UPC	1.03 251230	Miteq
	Peak-UPC7002	1.09 240418	Peak Communications
	Uplink-Power-Control-2	1.02 171222	SatService
	Uplink-Power-Control	1.12 241126	SatService
UPS	BMS-UPS	1.01 180813	BMS-IOM
	Generex-CS121-UPS-Adapter	1.01 171213	Generex
	MIB2-UPS-Monitor	1.04 250428	Generic
Variable-Power-Combiner	Mirad-VPC	1.03 171213	Mirad Microwave
	Mitec-Switch	1.02 171213	Mitec
Video-Converter	Ateme-TitanEdge-Bridge	1.02 251203	Ateme

Tag	*Driver*	*Version*	*Vendor*
Weather	ASCSignal-NGC-Weather	1.00 190725	ASCSignal
	ColumbiaWeather-MicroServer	1.00 250714	Vaisala
	Mesa-WNT	1.02 180223	Mesa
	Mesa-WSWD	1.02 180223	Mesa
	Vaisala-WXT	1.07 200916	Vaisala
	Wilmers-blueberryCOMPACT	1.05 180223	Wilmers

9.4 Logical Devices Overview

The sat-nms software contains a set so called logical devices, which in fact are software solution to solve different task in an M&C system.

The following list provides an overview about all these devices and the link to a detailed description.

- Antenna-Pointing --- A 'high level' driver for motor driven antennas. It is used together with a driver for the type of antenna controller used.
- Antenna-TrackingA 'high level' driver for motor driven antennas combined with step track software.
- Band-Pol-4-To-1 --- Controls a 4-to-1 switch which typically selects a receiver input signal from two polarization panes and two frequency bands.
- Channel-Select --- Channel Database
- Choice-Translator --- Provides a 2-way translation table for choice parameters.
- Device-CopyLets the operator copy the settings from one device to another with a single mouse click.
- Device-Mirror --- Ties two 1:1 redundant devices together. Any time a setting is changed at the primary device, the Device-Mirror copies this setting to the spare device.
- Dual-Translator --- Translates a set of user definable choices into the position settings of two switches (or equivalent). It may be used to combine two switches to a single function in the user interface.
- Fault-Array --- Combines fault messages or other state parameters of arbitrary devices in a flexible way to one common fault state.
- File-Recorder --- Provides a simple facility to write measurement or state values in regular intervals to a file.
- Frequency-Split --- Splits up a frequency value in a coarse part (1 MHz resolution) and a

fine part.

- Gate-Array --- Provides 8 logical gates with 8 inputs each. May be used to generate changing colors of user user interface elements due to state changes.
- IRD-Chain --- Controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization
- Level-Control --- Provides a gain setting linearizer for TWT amplifiers as well as a level control loop mode.
- Level-Set --- Provides a simple mechanism to set (and monitor) transmit power values in EIRP
- Link --- Connects the equipment used in a VLC for one end point of a satellite link to the link management of the NMS software. It controls the timing when links are going up or down and does all equipment settings necessary for the satellite link.
- Logger --- Watches an arbitrary parameter and, writes a message to the event log every time the parameter value changes.
- Logical-Gate-3 --- Provides a 3-input logical gate which can be used to merge status variables in a user programmed way.
- Logical-Gate --- Provides a simple 2-input logical gate which can be used to merge status variables in a user programmed way.
- Macro-Player --- Plays a macro if the value of a watched parameter changes in a certain way.
- Modcod-Splitter --- Makes a combined MODCOD parameter controllable through individual DVB-mode, modulation-type and FEC parameters.
- Parameter-Mirror --- Ties a single parameter of two 1:1 redundant devices together. Any time the parameter is changed at the primary device, the Parameter-Mirror copies this setting to the spare device.
- Pointing-Lock --- Checks the lock state of up to twenty IRD-Chain devices and controls the enable state of one Antenna-Pointing device.
- Protection-Switch-1-To-1 --- Provides a completely software based solution for a 1:1 redundancy switch. The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch the (usually transmit-) chain.
- Protection-Switch-2-To-1 --- Provides a completely software based solution for a 1:1 redundancy switch. The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch the (usually transmit-) chain. This device is contained for compatibility to existing sat-nms installations. For new installations the N:1 redundancy switch described below should be used.
- Protection-Switch-N-To-1 --- Provides a completely software based solution for a N:1 redundancy switch for transmit chains. Supersedes the Protection-Switch-2-To-1 device.
- Proxy-Device --- Maps a device from a remote sat-nms M&C onto the local system.
- Remote-Alarm --- The Remote-Alarm logical device is part of the remote alarm client feature of the software which reports failures of a M&C system to a remote site through a ('on demand') dial up connection.
- Replicator --- Replicates a parameter setting from one device to another.
- RX-ChannelSelect --- Applies tuning parameters and input signal selection from a common channel database to an IRD, including LO frequency calculations for multiple antennas.
- RX-ChannelSelect-CDB --- Similar to RX-ChannelSelect, but for usage in an NMS environment, using a central database (CDB) located in the NMS server for the channel lists.

- SNG-Pointing --- Adds some useful functions to the basic control of car mounted antennas.
- Site-Switch --- Implements a site diversity switch control and is mostly used to backup up-link facilities at different sites.
- Switch-Cross-Link --- Translates a ON/OFF command into a set of arbitrary messages / commands.
- Sync-Switch --- Couples two switches located at two different site to one single switch.
- Transmit-Chain-Preset --- Manages to store / retrieve the settings of a complete transmit chain in one preset. The device collects the setting of a polarization switch, the frequency and the EIRP of the transmission and the device presets of two arbitrary devices (e.g. modulator and encoder) into one DevicePreset.
- TX-ChannelSelect --- Applies tuning parameters from a common channel database to an encoder / modulator.
- TX-ChannelSelect-CDB --- Similar to TX-ChannelSelect, but for usage in an NMS environment, using a central database (CDB) located in the NMS server for the channel lists.

9.5 ACTi-ACM1231-Webcam

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.userName	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
videoLAN.cmds	TEXT	R/O	StringRange R/O
address	TEXT	SAVE	StringRange
brightness	INTEGER		IntegerRange (1 .. 100)

name	*type*	*flags*	*range*
contrast	INTEGER		IntegerRange (1 .. 100)
saturation	INTEGER		IntegerRange (1 .. 100)
resolution	CHOICE		EnumRange (160x112 320x240 640x480 1280x720 1280x1024)
resetStreaming	TEXT	NOPRESET	StringRange

9.6 APC-ATS

This page describes the device driver and the device window for APC Automatic Transfer Switches via SNMP.

The driver was tested with the following models:

- AP4421 SW: 4.34
- AP7723 SW: 3.05

NOTE: Please enable SNMP access (v1) and set read/write community via WebUI

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- frontPanel --- Defines number of available channels
- voltageSensitivity --- Defines units voltage sensitivity HIGH/LOW
- voltageRange --- Defines units voltage range WIDE/MEDIUM/NARROW
- currentLimit --- Defines units current limit (0..20A)
- lineVrms --- Defines line VRMS in V
- wideVrms --- Defines wide VRMS in V
- mediumVrms --- Defines medium VRMS in V
- narrowVrms --- Defines narrow VRMS in V
- freqDeviation --- Defines frequency deviation in Hz
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.nomVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.nomFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.inPhases	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outPhases	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.phaseSync	CHOICE	R/O	EnumRange R/O (SYNC OUT OF SYNC)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.frontPanel	CHOICE	SETUP	EnumRange (ENABLE DISABLE)
config.voltageSensivity	CHOICE	SETUP	EnumRange (HIGH LOW)
config.voltageRange	CHOICE	SETUP	EnumRange (WIDE MEDIUM NARROW)
config.currentLimit	INTEGER	SETUP	IntegerRange (0 .. 20)
config.lineVrms	INTEGER	SETUP	IntegerRange (207 .. 253)
config.wideVrms	INTEGER	SETUP	IntegerRange (3 .. 35)
config.mediumVrms	INTEGER	SETUP	IntegerRange (3 .. 35)
config.narrowVrms	INTEGER	SETUP	IntegerRange (3 .. 35)
config.freqDeviation	CHOICE	SETUP	EnumRange (1Hz 2Hz 3Hz 5Hz 10Hz)
config.deviceComment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
priority	CHOICE		EnumRange (A B)
position	CHOICE	R/O	EnumRange R/O (A B)
load.highThreshold	INTEGER		IntegerRange (0 .. 0)
load.highWarning	INTEGER		IntegerRange (0 .. 0)
load.lowWarning	INTEGER		IntegerRange (0 .. 0)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.01.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.01.voltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.voltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.voltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.load	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.loadPercent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.loadState	CHOICE	R/O	EnumRange R/O (NORMAL LOW-LOAD NEAR-OVERLOAD OVERLOAD)
reset	TEXT	NOPRESET	StringRange
clearFaults	TEXT	NOPRESET	StringRange
internal.model	TEXT	R/O	StringRange R/O
internal.model2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Internal Comms)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Redundancy)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Over current)
faults.04	ALARM	R/O	AlarmFlagRange R/O (5V PSU)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (24V PSU A)
faults.06	ALARM	R/O	AlarmFlagRange R/O (24V PSU B)
faults.07	ALARM	R/O	AlarmFlagRange R/O (+12V PSU)
faults.08	ALARM	R/O	AlarmFlagRange R/O (-12V PSU)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Source A)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Source B)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Voltage Out)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.15	ALARM	R/O	AlarmFlagRange R/O (3.3V PSU)
faults.16	ALARM	R/O	AlarmFlagRange R/O (1.0V PSU)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Boost Voltage Source A)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Boost Voltage Source B)

9.7 APC-RPDU

This page describes the device driver and the device window for APC rPDU series PowerModules via SNMP.

The driver was tested with the following models:

- AP7920

NOTE: Please set SNMP write community access rights to "write+" within WebUI of APC to allow on/off/reboot

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- pulseTime.01 --- Defines the pulse time in sec for outlet 1
- pulseTime.02 --- Defines the pulse time in sec for outlet 2
- pulseTime.03 --- Defines the pulse time in sec for outlet 3
- pulseTime.04 --- Defines the pulse time in sec for outlet 4
- pulseTime.05 --- Defines the pulse time in sec for outlet 5
- pulseTime.06 --- Defines the pulse time in sec for outlet 6
- pulseTime.07 --- Defines the pulse time in sec for outlet 7
- pulseTime.08 --- Defines the pulse time in sec for outlet 8

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.outlets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.pulseTime.01	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.02	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.03	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.04	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.05	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.06	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.07	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
config.pulseTime.08	INTEGER	SETUP	IntegerRange DIS (5 .. 60)
name.01	TEXT		StringRange DIS
name.02	TEXT		StringRange DIS
name.03	TEXT		StringRange DIS
name.04	TEXT		StringRange DIS

name	*type*	*flags*	*range*
name.05	TEXT		StringRange DIS
name.06	TEXT		StringRange DIS
name.07	TEXT		StringRange DIS
name.08	TEXT		StringRange DIS
state.01	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.02	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.03	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.04	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.05	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.06	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.07	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
state.08	CHOICE	R/O	EnumRange R/O DIS (ON OFF)
control.01	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.02	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.03	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.04	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.05	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.06	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)

name	*type*	*flags*	*range*
control.07	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
control.08	CHOICE	NOPRESET NOCOMPARE	EnumRange DIS (ON OFF REBOOT DELAYED-ON DELAYED-OFF DELAYED-REBOOT CANCEL)
meas.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Too many outlets)

9.8 APS-Backup-Antenna-Control

The APS-Backup-Antenna-Control logical device is a project specific extension to the sat-nms software. The device controls the switch over procedure between a backup antenna and a number of operative antennas.

Switching the uplink from an operative antenna to the backup antenna is a 2-step procedure. The first step (preparation) positions the backup antenna to the same satellite as the operative antenna and prepares the waveguide switches at the backup antenna for taking over the signal(s) from the operative antenna.

After the preparation has been completed successfully the final switchover is executed as an independent command. The actual switchover is performed by a APS-Backup-Antenna-Switch logical device located at the operative antenna. This sets the waveguide switches there to route the uplink to the backup antenna instead to the local one. Immediately after the switchover this device also increases the transmit power to compensate for the increased waveguide loss to the backup antenna. At the backup antenna M&C, Proxy devices act in place of the remote APS-Backup-Antenna-Switch devices.

Antenna command and verification

As of version 2.00 the APS-Backup-Antenna-Control device expects the backup antenna to be controlled by a SatService ACU/ODM (SatService-ACU-ODM device driver). During the switchover preparation phase, the APS-Backup-Antenna-Control device commands a target memory number to the ACU/ODM. The target memory in the ACU contains the center of box coordinates of the satellite and the beacon/tracking parameters stored for the satellite normally tracked by the antenna which is going to be backed up.

The center of box coordinates are stored in the APS-Backup-Antenna-Control device's configuration file as well. After the the antenna has been commanded to move to the target in question, the APS-Backup-Antenna-Control device polls the actual antenna pointing and compared it to the pointing stored in the configuration file. Once the pointing matches within a

configurable threshold, the APS-Backup-Antenna-Control device states "position reached" and proceeds to the next step.

With this next step the beacon level threshold fault is monitored. The device waits until there is no beacon level stated by the ACU, then the device assumes the antenna points to the satellite.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the signal destination setting.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

Most of the configuration parameters of the APS-Backup-Antenna-Control device are defined in a configuration file (see below). At the maintenance page of the device window there are a couple of configuration parameter which concern the local (backup) antenna device.

parameter	***description***
antennaName	The device name of the backup antenna controller. This device must be a NDSatCom-Step-Track device.
antennaTimeout	The maximum time, the APS-Backup-Antenna-Control device waits after an antenna movement, that the antenna tracks the new satellite.
azThreshold	The difference between measured and commanded antenna pointing accepted by the APS-Backup-Antenna-Control device to state 'position reached'.
elThreshold	The difference between measured and commanded antenna pointing accepted by the APS-Backup-Antenna-Control device to state 'position reached'.

Fault flags

The device provides three fault flags:

- The 'Configuration file FAULT' is set if the device could not parse it's configuration file on startup.
- The 'Switch combination FAULT' is set if the combination of waveguide switch settings does not match any of the configured combinations. Command a 'prepare', optionally followed by a 'release' to reset this fault.

- The 'Switchover procedure FAULT' is set if a fault occurs during a prepare or switchover procedure. The 'reset' command clears this fault.

Configuration file

The APS-Backup-Antenna-Control device requires to be setup by a configuration file. The file has the name of the device (case sensitive!) extended by the suffix '.txt'. The file must reside in the M&C server's home directory. It defines the number of stations this antenna shall backup, the antenna pointings and all switch setting to be performed during the preparation / switchover.

The file format text based, straight forward and almost self-explaining. This is a short example, taken from the real installation in Munich. Everything right of a '#' character is treated as a comment, in the example below those are used used to describe the file format.

```
#
#  Example for an APS-Backup-Antenna-Control configuration file
#
#
   S1-2X = A    # WG switch setting for the local operation
   S1-2Y = A    # of the backup antenna. The 'forceLocal' command
   S1-3X = A    # tells the APS-Backup-Antenna-Control to perform
   S1-3Y = A    # these settings
   S2-1X = A
   S2-1Y = A
   S1-AX = B
   S1-AY = B
   S2-AX = B
   S2-AY = B

#  The 'station' statement opens the definition of a antenna / M&C to backup.
#  '1A' is the name of the antenna which appears in the selection box a the
#  APS-Backup-Antenna-Control's user interface.
#
station 1A

   S1-2X = B    # The WG switch settings following the 'station' statement
   S1-2Y = B    # are performed when the operator commands to prepare the
   S1-3X = B    # take over for this antenna. They are performed immediately,
   S1-3Y = B    # before the antenna gets moved.

   # The 'antenna' statement tells the device where to point the antenna. It
   # must be followed by the target memory number to be invoked at the
   # antenna controller and the azimuth / elevation reference angles on
   # the same line.
   #
   antenna  2  169.958  34.115
```

```
    SU-4X = B    # After the antenna has been pointed to the new satellite and
    SU-4Y = B    # tracking has found the satellite, the WG switch settings
    SU-3X = B    # following the 'antenna' statement are performed. After
    SU-3Y = B    # these settings have been verified successfully, the
    S1-AX = A    # backup antenna is prepared for the switchover.
    S1-AY = A

    # The 'final' statement closes the definition opened by 'station'. It
    # tells the APS-Backup-Antenna-Control device which proxy device
    # has to be controlled in order to execute the actual switchover at the
    # remote antenna. The proxy device must be configured to act for the
    # APS-Backup-Antenna-Switch device at the M&C of interrest.
    #
    final ASTRA-1A

#  Here starts the definition of the second station (omitted here). The
#  APS-Backup-Antenna-Control device is capable to manage an arbitrary number
#  of stations.
#
station 1B
```

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.antennaName	TEXT	SAVE SETUP	StringRange
config.antennaTimeout	INTEGER	SAVE SETUP	IntegerRange (1 .. 15)
config.azThreshold	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 1.000)
config.elThreshold	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 1.000)
station	CHOICE		EnumRange (??)
prepare	TEXT		StringRange DIS
release	TEXT		StringRange DIS

name	*type*	*flags*	*range*
switchTo	TEXT		StringRange DIS
switchBack	TEXT		StringRange DIS
forceLocal	TEXT		StringRange
progressReport	TEXT	R/O	StringRange R/O
progressIcon	TEXT	R/O	StringRange R/O
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Configuration file)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch combination)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switchover procedure)

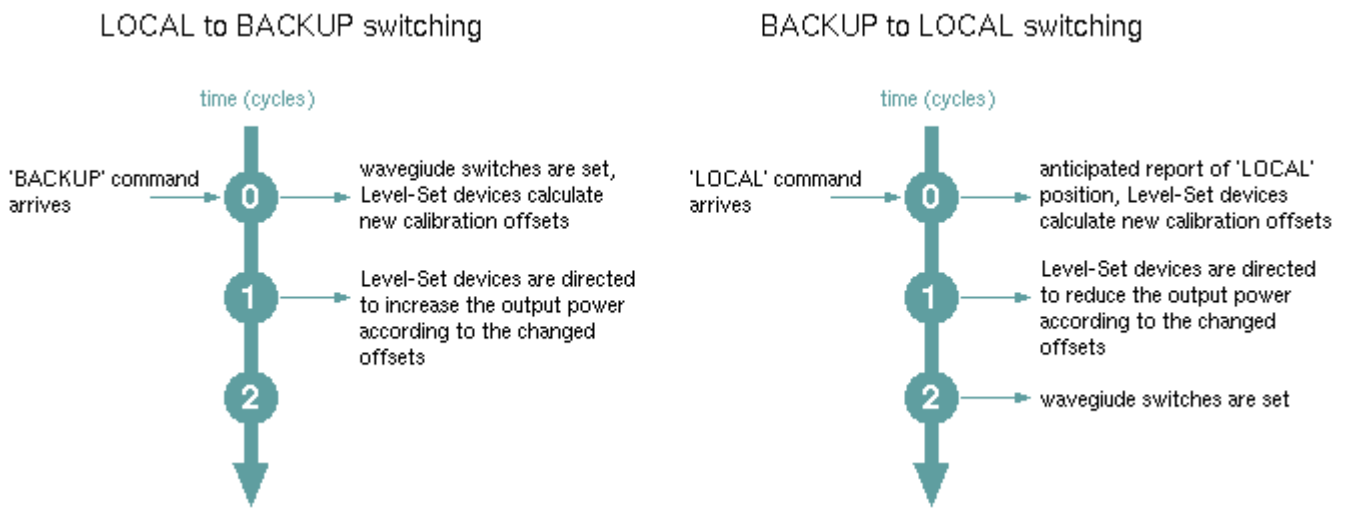
9.9 APS-Backup-Antenna-Switch

The APS-Backup-Antenna-Switch logical device is a project specific extension to the sat-nms software. The device controls the signal path to from the HPAs / power combiners to the antenna. For this the device commands a number of wave guide switches. The device names of the switches are configurable as the individual switch positions are as well. The APS-Backup-Antenna-Switch device is used in conjunction with the APS-Backup-Antenna-Control device which manages the sequence of operations required to switch an antenna's transmit signal to the backup antenna.

The APS-Backup-Antenna-Switch provides three signal destination choices:

- **LOCAL** --- The transmit signal gets routed to the local antenna.
- **BACKUP** --- The transmit signal gets routed to the backup antenna. The wave guide switches at the backup antenna *must* have been set to the proper positions before.
- **DUMMY** --- The transmit signal gets routed to a dummy load. It depends on the configuration of the APS-Backup-Antenna-Switch if this destination selection is available for a particular antenna.

One task of the APS-Backup-Antenna-Switch device is to manage the individual waveguide switch settings necessary for the procedure to switch to the backup antenna and back up the local one. The other task of the device is to let the Level-Set devices attached to each HPA know about the actual signal path and to command them to adjust the power at the right time. The sequence of operations is as shown in the diagram below:



When switching to BACKUP (requires a higher power at the HPAs due to the longer waveguide path), the device first sets the waveguide switches and reports the signal destination to be BACKUP. The Level-Set devices monitoring this value calculate new calibration offsets due to the change, but do not change the ouptu power. One cycle (usually 1000 msesc) later, when the switches are in the BACKUP position, the APS-Backup-Antenna-Switch commands the Level-Set devices to increase the output power according to the changes calibration values.

When switching to LOCAL (this requires to reduce the output power), the device first reports the LOCAL setting although the waveguide switches have not yet been set. This causes the Level-Set device calculate the calibration offsets for the LOCAL signal path. One cycle later the output power reduction is commanded. In a third cycle the APS-Backup-Antenna-Switch device sets the waveguide switches to the LOCAL.

Switching the signal destination to DUMMY (dummy load) does not cause a change of the amplifiers' output power remains.

Destination monitoring

At power on, the APS-Backup-Antenna-Switch detects the actual destination selection from the waveguide switch positions (they are the 'permanent' memory for this parameter). If the combination of the switch positions is not valid (does not match any of the switch combinations defined with the configuration parameters), a fault flag is set.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the signal destination setting.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the switches that will perform the signal routing.

parameter	***description***
lsetNames	A comma separated list of devices names. Each device must be a Level-Set device which shall receive an 'adjustDrive' command during the switchover sequence as described above. This list may be empty if there are no Level-Set devices to control.
Example: **EIRP-1,EIRP-2,EIRP-3,EIRP-R**	
swNames	A comma separated list of devices names. Each device must be a 'switch' type in that way, that is provides a variable called 'position' which selects the signal path.
Example: **S01,S02,S03,S04**	
swLocal	A comma separated list defining the switch commands (positions) which get applied when the device switches to the LOCAL position. The number of items in the list must match the number of switches defined above. Switches and position commands are linked by their position in the list.
Example: **A,A,B,B**	
When switching to LOCAL, S01 and S02 are set to position 'A', S03 and S04 are set to 'B' following this example.	
swBackup	A comma separated list defining the switch commands (positions) which get applied when the device switches to the BACKUP position. The number of items in the list must match the number of switches defined above. Switches and position commands are linked by their position in the list.
Example: **B,B,A,A**	
When switching to BACKUP, S01 and S02 are set to position 'B', S03 and S04 are set to 'A' following this example.	

parameter	***description***
swDummy	A comma separated list defining the switch commands (positions) which get applied when the device switches to the DUMMY position. The number of items in the list must match the number of switches defined above or the field must be left empty. Switches and position commands are linked by their position in the list.
Example: **A,B,A,B**	
When switching to DUMMY, S01 and S03 are set to position 'A', S02 and S04 are set to 'B' following this example. If the field is left empty, the devices suppresses the DUMMY destination.	

Changes of the configuration parameter are recognized immediately. If this temporarily leads to invalid switch combinations, the 'Switch position match FAULT' is set. It may be necessary to switch the 'position' parameter once in order to clear this fault.

Fault flags

The device provides two fault flags:

- The 'Switch position match FAULT' is set if the actual positions of the waveguide switches do not match any of the combinations configured for this device. This may happen with a defective switch or if a switch is controlled manually. The fault disappears when the switch position gets fixed.
- The 'Switch hardware FAULT' is set if at least one of the switch devices has it's summary fault set.

Configuration hints

To make the APS-Backup-Antenna-Switch device interact properly with a Level-Set device, both devices mut be configured as follows:

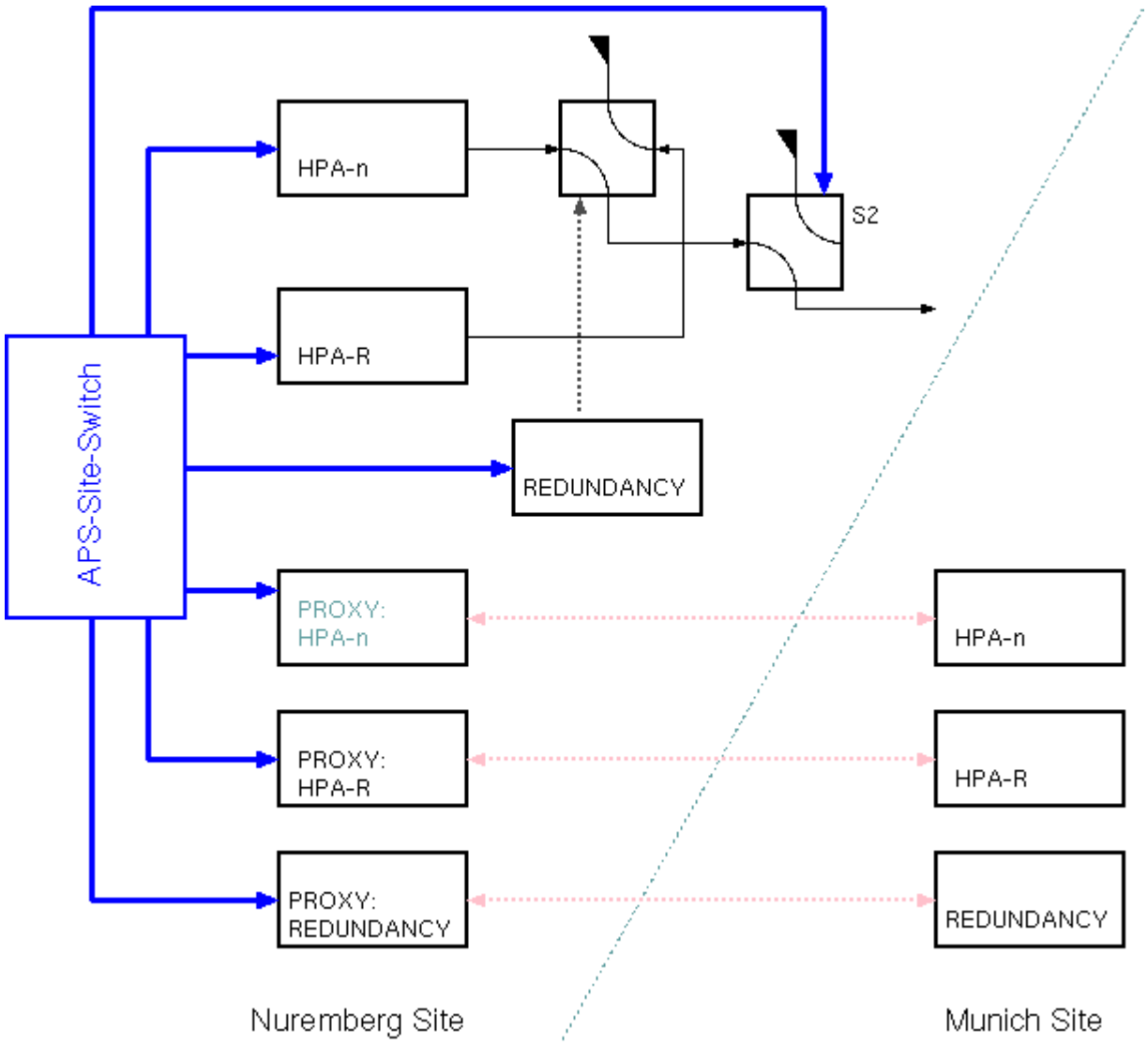
- The Level-Set device must have the 'position' variable of the APS-Backup-Antenna-Switch set at either the 'txSwitch1Id' or 'txSwitch2Id' configuration setting in order to monitor the selected signal path. The Level-Set device must not monitor the waveguide switch positions directly!
- At the APS-Backup-Antenna-Switch, the name of the Level-Set device must appear in the 'lsetNames' configuration setting. This makes the APS-Backup-Antenna-Switch sending 'adjustDrive' commands to the Level-Set device during the switchover sequence.
- At the Level-Set device, the 'EIRP low/high threshold' faults should be configured with a 5 seconds delay to avoid unwanted fault messages during the switchover.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.lsetNames	TEXT	SAVE SETUP	StringRange
config.swNames	TEXT	SAVE SETUP	StringRange
config.swLocal	TEXT	SAVE SETUP	StringRange
config.swBackup	TEXT	SAVE SETUP	StringRange
config.swDummy	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (LOCAL BACKUP)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Switch position match)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch hardware)

9.10 APS-Site-Switch

The APS-Site-Switch logical device is a project specific extension to the sat-nms software. The device controls the equipment of one transmit chain for the site diversity application between Nuremberg and Munich.



As shown in the figure above, the APS-Site-Switch logical device is operated at the Nuremberg site. There it controls the HPA which shall take over the Munich uplink, the redundant HPA, the protection switch logical device and the waveguide switch S2 which routes the HPA output either to dummy load or to the antenna. Beside this, the APS-Site-Switch controls three Proxy devices which must be configured to resemble the corresponding HPAs and the protection switch at the Munich site.

Site switch procedure

Switching the uplink from Munich to Nuremberg is a 2-step procedure. With the first step, Nuremberg gets prepared to take over the uplink, the real site switch then is performed as the second step.

To prepare Nuremberg to take over the uplink, the function 'prepare' performs the following operations:

1. Switch off the HPA protection / redundancy switch.
2. Detect if HPA-? or the redundant one must be used.

3. Switch the HPA of interest to RF-OFF.
4. Poll the HPA up to 10 seconds until it states RF-OFF. Abort on failure.
5. Switch the HPA BEAM-ON
6. Switch S2 to antenna.

The Nuremberg transmit facility is signalled to be prepared, when the APS-Site-Switch detects the protection switch to be OFF, the HPA to be RF-OFF / BEAM-ON and S2 set to antenna.

The actual switch over can only be performed, if the 'prepared' condition is detected by the APS-Site-Switch. This enables the 'NBG' button at the user interface. The steps of the switch over are:

1. Switch OFF protection for the Munich HPA, remember the value to restore it when switching back.
2. Detect if HPA-? or the redundant one must be used at Munich.
3. Switch the Munich HPA to RF-OFF.
4. Poll the HPA up to 10 seconds until it states RF-OFF.
5. Switch the Nuremberg HPA to RF-ON.

Switching back to Munich works the same way the other way round.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands actual uplink location.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the devices that will perform the carrier on/off settings.

parameter	***description***
s2Device	The device name of the waveguide switch which routes the local uplink either to the antenna or to a dummy load.
s2Positions	Two switch positions for the S2 switch, separated by a comma. The first value is used to switch S2 to dummy load, the second one to switch it to antenna.
lclHpaDevice	The device name of the local HPA.
lclProtectDevice	The device name of the local protection switch device.

parameter	*description*
rmtHpaDevice	The device name of the Proxy device which resembles the corresponding HPA at the Munich site.
rmtProtectDevice	The device name of the proxy device which resembles redundancy switch at the Munich site.
comment	A short text which is displayed on the main page of the device window to ease the identification of the transmit chain.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.s2Device	TEXT	SAVE SETUP	StringRange
config.s2Positions	TEXT	SAVE SETUP	StringRange
config.lclHpaDevice	TEXT	SAVE SETUP	StringRange
config.lclProtectDevice	TEXT	SAVE SETUP	StringRange
config.rmtHpaDevice	TEXT	SAVE SETUP	StringRange
config.rmtProtectDevice	TEXT	SAVE SETUP	StringRange
config.comment	TEXT	SAVE SETUP	StringRange
muc	TEXT		StringRange
nbg	TEXT		StringRange
prepare	TEXT		StringRange
release	TEXT		StringRange
progressReport	TEXT	R/O	StringRange R/O
progressIcon	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
reset	TEXT		StringRange
mucAll	TEXT		StringRange
nbgAll	TEXT		StringRange
prepareAll	TEXT		StringRange
releaseAll	TEXT		StringRange

9.11 ARD-Backup-Client

The *ARD-Backup-Client* is a customer specific logical device which controls a backup transmit chain located an another M&C system using a proprietary UDP based remote control protocol. The remote backup transmit chain may be controlled by a third party M&C system implementing this protocol or by a sat-nms M&C using the ARD-Backup-Server logical device.

The *ARD-Backup-Client* shows some state information about the remote transmit chain in it's device window and provides buttons to switch this chain to antenna or to load. Beside this, buttons are provided to increment or decrement the chain's transmit power in steps of 1 or 0.5 dB.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Monitoring an control of the remote backup transmit chain
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	*description*
chainName	The name of the backup transmit chain at the remote M&C. This name is used in the communication protocol to address the chain to be controlled. Typical values are 'SZ1' or 'SZ2'

parameter	*description*
listenPort	The port number at which the ARD-Backup-Client listens for status messages from the remote M&C. This port number must match the 'clientPort' set at it's counterpart.
serverPort	The port number to which the ARD-Backup-Client sends any command messages. This port must match the 'listenPort' set at the server side.
serverIP	The IP address of the remote M&C.
key	The key used by the communication protocol to encrypt/decrypt messages. The key is a 256 digit hexadecimal string. Normally there is no need to change this parameter.
wgsId	The device name of the wave guide switch used to switch the signal either to antenna or to load. The ARD-Backup-Client expects this device to provide a 'position' parameter which controls the switch position.
wgsXlate	A comma separated list of two switch positions. The first position indicates 'LOAD', the second one 'ANTENNA'. If for example the switch routes the signal to antenna for position 'A', wgsXlate must be set to 'B,A'.
powerDetectId	The ID of a parameter which indicates if there is transmit power detected after the wave guide switch. The ARD-Backup-Client expect this to be a boolean type value with 'true' indicating the presence of transmit power at the antenna.

Fault flags

The device provides a 'faults.99' fault flag which indicates that the communication to the remote M&C is broken. The flag is raised if there was no valid status message received for 5 seconds.

Remarks

The device uses it's device thread to detect communication timeouts. For this reason, the device should be configured as the only device at a 'null' interface with an idle time of 1000 msecs.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.chainName	TEXT	SAVE SETUP	StringRange
config.listenPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.serverPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.serverIP	TEXT	SAVE SETUP	StringRange
config.localIP	TEXT	SAVE SETUP	StringRange
config.key	TEXT	SAVE SETUP	StringRange
config.wgsId	TEXT	SAVE SETUP	StringRange
config.wgsXlate	TEXT	SAVE SETUP	StringRange
config.powerDetectId	TEXT	SAVE SETUP	StringRange
pwrInc05	TEXT	NOPRESET	StringRange
pwrInc10	TEXT	NOPRESET	StringRange
pwrDec05	TEXT	NOPRESET	StringRange
pwrDec10	TEXT	NOPRESET	StringRange
toRemote	TEXT	NOPRESET	StringRange
toLocal	TEXT	NOPRESET	StringRange
state.name	TEXT	R/O	StringRange R/O
state.wgs	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.wgsText	TEXT	R/O	StringRange R/O
state.upconverter	TEXT	R/O	StringRange R/O
state.modulator	TEXT	R/O	StringRange R/O
state.hpa	TEXT	R/O	StringRange R/O
state.powerDetect	BOOLEAN	R/O	BooleanRange R/O (true false)
state.hpaPower	FLOAT	R/O	DoubleRange R/O (0 .. 0)
state.modCod	TEXT	R/O	StringRange R/O
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.asiMode	TEXT	R/O	StringRange R/O
state.eirp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.eirpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.summaryFault	TEXT	R/O	StringRange R/O
state.operationLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
state.message	TEXT	R/O	StringRange R/O
state.toggle	BOOLEAN	R/O	BooleanRange R/O (true false)
state.rawmsg	TEXT	R/O	StringRange R/O
localWgs	TEXT	R/O	StringRange R/O
remoteWgs	TEXT	R/O	StringRange R/O
localPowerDetect	TEXT	R/O	StringRange R/O
remotePowerDetect	TEXT	R/O	StringRange R/O

9.12 ARD-Backup-Server

The *ARD-Backup-Server* is a customer specific logical device which monitors and controls some

parameters of a backup transmit chain and permits remote control of these parameters from another M&C system using a proprietary UDP based remote control protocol. The remote M&C may be a third party M&C system implementing this protocol or by a sat-nms M&C using the ARD-Backup-Client logical device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows some status information.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
transponder	The name of the transponder this backup transmit chain is red for. The transponder name is part of the communication protocol and displayed as a header at the client GUI.
chainName	The name of the backup transmit chain. This name is used in the communication protocol to address the chain to be controlled. Typical values are 'SZ1' or 'SZ2'
dotlistenPort	The port number at which the ARD-Backup-Server listens for command messages from the remote M&C. This port number must match the 'serverPort' set at it's counterpart.
clientPort	The port number to which the ARD-Backup-Server sends it's status messages. This port must match the 'listenPort' set at the client side.
clientIP	The IP address of the remote M&C.
key	The key used by the communication protocol to encrypt/decrypt messages. The key is a 256 digit hexadecimal string. Normaly there is no need to change this parameter.
asiInputId	The parameter ID of the modulator's ASI input selection. The ARD-Backup-Server expects this parameter to one of '1'. '2'. If the referenced modulator parameter uses other values, a Choice-Translator device should be used to adapt the selections.

parameter	***description***
wgsId	The device name of the wave guide switch used to switch the signal either to antenna or to load. The ARD-Backup-Server expects this device to provide a 'position' parameter which controls the switch position.
wgsXlate	A comma separated list of two switch positions. The first position indicates 'LOAD', the second one 'ANTENNA'. If for example the switch routes the signal to antenna for position 'A', wgsXlate must be set to 'B,A'.
upconverterId	The device name of the upconverter in the chain.
modulatorId	The device name of the modulator in the chain.
hpaId	The device name of the HPA in the chain.
powerDetectId	The ID of a parameter which indicates if there is transmit power detected after the wave guide switch. The ARD-Backup-Server expect this to be a boolean type value with 'true' indicating the presence of transmit power at the antenna.
hpaPowerId	The parameter ID reporting the HPA output power. The referenced parameter should report the HPA output power in Watts.
dvbModeId	The parameter ID reporting the modulator's DVB-S / DVB-S2 setting. For devices using combined modcod settings see 'Remarks' below.
modulationId	The parameter ID reporting the modulator's modulation (QPSK, 8PSK etc.) setting. For devices using combined modcod settings see 'Remarks' below.
fecId	The parameter ID reporting the modulator's FEC setting. For devices using combined modcod settings see 'Remarks' below.
frequencyId	The parameter ID reporting the chain's transmit frequency (MHz). The parameter should report the real TX frequency at the antenna, no IF or L-Band frequency.
symbolRateId	The parameter ID reporting the modulator's symbol rate (Msym/sec)
dataRateId	The parameter ID reporting the modulator's interface data rate (MBit/s)
asiModeId	The parameter ID reporting the modulator's ASI mode setting (packet size)
eirpId	The ID of the parameter reporting the (measured) transmitted EIRP (dBW).

parameter	*description*
eirpFaultId	The parameter ID of a fault flag indicating that the measured EIRP values is outside it's configured limits.
pwrControlId	The ID of the parameter controlling the transmit power. The value should be dB-based, may refer to either a gain setting or to an attenuation.
pwrControlMode	Selects if powerControlId refers to an 'ATTEN' or to a 'GAIN' parameter, thus reverting the sense of the power increment buttons as required.

Remarks

The ARD-Backup-Server device sends status messages to the client on a regular basis. The device sends one message per driver cycle. Hence it is recommended to configure the device as the only one at a 'null' device. The IDLE time setting of this interface lets you control the frequency of the status message transmissions.

The device combines the state information received through the 'dvbModeId', 'modulationId' and 'fecId' parameters to a combined modcod string looking like 'DVBS-QPSK-3/4'. If the modulator already uses combined modcod settings, you may use such a parameter in place of the individual settings. If for instance the modulator used a combined modulation-fec setting, you may set this parameter at the 'modulationId' an leave the 'fecId' setting empty.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.transponder	TEXT	SAVE SETUP	StringRange
config.chainName	TEXT	SAVE SETUP	StringRange
config.listenPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.clientPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.clientIP	TEXT	SAVE SETUP	StringRange
config.localIP	TEXT	SAVE SETUP	StringRange
config.key	TEXT	SAVE SETUP	StringRange
config.asiInputId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.wgsId	TEXT	SAVE SETUP	StringRange
config.wgsXlate	TEXT	SAVE SETUP	StringRange
config.upconverterId	TEXT	SAVE SETUP	StringRange
config.modulatorId	TEXT	SAVE SETUP	StringRange
config.hpaId	TEXT	SAVE SETUP	StringRange
config.powerDetectId	TEXT	SAVE SETUP	StringRange
config.hpaPowerId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dataRateId	TEXT	SAVE SETUP	StringRange
config.asiModeId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.eirpFaultId	TEXT	SAVE SETUP	StringRange
config.pwrControlId	TEXT	SAVE SETUP	StringRange
config.pwrControlMode	CHOICE	SAVE SETUP	EnumRange (ATTEN GAIN)
state.control	TEXT	R/O	StringRange R/O
state.wgs	TEXT	R/O	StringRange R/O
state.upconverter	TEXT	R/O	StringRange R/O
state.modulator	TEXT	R/O	StringRange R/O
state.hpa	TEXT	R/O	StringRange R/O
state.message	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fault 1)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fault 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fault 3)

9.13 ASCSignal-NGC-ACU

This page describes the device driver and the device window for the ASC Signal NGC ACU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the tracking parameters.
- --- Contains the monopulse tracking parameters if applicable.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- jogRate --- The speed used to jog the antenna (0-100%)

- jogRate --- The speed used to jog the antenna (0-100%)
- monopulse --- Enables/disables the monopulse tracking parameters.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azLimits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.elLimits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.plLimits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lowSignal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.moving	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.jogRate	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)

name	*type*	*flags*	*range*
config.monopulse	CHOICE	SAVE SETUP	EnumRange (DISABLED ENABLED)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
jogMode	INTEGER	NOPRESET	IntegerRange (0 .. 0)
trackCommand	INTEGER	NOPRESET	IntegerRange (0 .. 0)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azMotor	TEXT	R/O	StringRange R/O
state.elMotor	TEXT	R/O	StringRange R/O
state.plMotor	TEXT	R/O	StringRange R/O
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.lvlSigma	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.stepDistance	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.satIndex	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.target	TEXT	R/O	StringRange R/O
trackMode	CHOICE		EnumRange (OFF STEP TRACK SMART ORBITAL NORAD INTELSAT ADAPTIVE STEP NORAD ADAPTIVE INTELSAT ADAPTIVE SMART 1ST ORDER TABLE TRACK MONOPULSE SMART TABLE TRACK SUN TRACK MOON TRACK)

name	*type*	*flags*	*range*
stepCycle	FLOAT		DoubleRange (0.0 .. 0.0)
stepThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
lowSignalThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
autoResume	CHOICE		EnumRange (OFF ON)
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.magnitude	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.angleError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mp.quietLevel	FLOAT		DoubleRange (0.0 .. 0.0)
mp.xelevationSatLevel	FLOAT		DoubleRange (0.00 .. 0.00)
mp.elevationSatLevel	FLOAT		DoubleRange (0.00 .. 0.00)
mp.dampingNumerator	INTEGER		IntegerRange (0 .. 0)
mp.dampingDenominator	INTEGER		IntegerRange (0 .. 0)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ hi hard limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ lo hard limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ hi soft limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ lo soft limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ hi box limit)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ lo box limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EL hi hard limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (EL lo hard limit)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL hi soft limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL lo soft limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL hi box limit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (EL lo box limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PL hi hard limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PL lo hard limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL hi soft limit)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PL lo soft limit)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Beacon level)

9.14 ASCSignal-NGC-BDC

This page describes the device driver and the device window for Work Dual Band BDC integrated in ASC Signal NGC 4000 ACU.

The driver was tested with the following FW:

- FW Work: FCA02.06

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- RX --- Shows the units state and gain/attenuation setting
- --- Shows the temperature measurent and reference source.
- --- Page not applicable for this device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- moduleIndex --- Sets the module index which applies for the BDC in the ACU system setup. Setting this parameter is indispensable for proper communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.moduleIndex	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.on.2	CHOICE	R/O	EnumRange R/O (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 20.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Master Standby)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel2 Standby)

9.15 ASCSignal-NGC-Deice

This page describes the device driver and the device window for the deicing integrated in ASCSignal NGC 400 ACU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements and subsystem states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.feedblowerOffDelay	FLOAT	SETUP	DoubleRange (2.0 .. 15.0)
config.sandblowerOnCycle	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sandblowerOffCycle	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.heaterZoneMode	CHOICE	SETUP	EnumRange (UNINITIALIZED DISABLED SINGLE ZONE MAT MULTI ZONE MAT SINGLE ZONE PLENUM MULTI ZONE PLENUM SINGLE ZONE FURNACE MULTI ZONE FURNACE EXTERNAL MULTI PLENUM DAMPERED)
config.heaterZoneStrategy	CHOICE	SETUP	EnumRange (UNITIALIZED NOT SET DEICE DEICE AND BALANCE SLAVE BALANCE)
config.heaterMaxTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.heaterZoneBalance	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refSetpoint.1	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refSetpoint.2	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refSetpoint.3	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refSetpoint.4	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refOffDelay.1	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.refOffDelay.2	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refOffDelay.3	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.refOffDelay.4	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.subreflector	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rain	BOOLEAN	R/O	BooleanRange R/O (true false)
blower	BOOLEAN	R/O	BooleanRange R/O (true false)
state.blower	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF OPERATIONAL FLT LOW PRESSURE CLOGGED BLOCKED SAND CYCLE PRESSURE SENSOR FLT)
blower.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heater.1	BOOLEAN	R/O	BooleanRange R/O (true false)
state.heater.1	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF BLOCKED)
heater.1.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heater.2	BOOLEAN	R/O	BooleanRange R/O (true false)
state.heater.2	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED OFF BLOCKED BALANCING)
deice.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.4	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.5	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
heater.ref.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.ref.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.ref.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.ref.4	BOOLEAN	R/O	BooleanRange R/O (true false)
state.heater.ref.1	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED OFF BLOCKED BALANCING)
state.heater.ref.2	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED OFF BLOCKED BALANCING)
state.heater.ref.3	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED OFF BLOCKED BALANCING)
state.heater.ref.4	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED OFF BLOCKED BALANCING)
heater.ref.1.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heater.ref.2.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heater.ref.3.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heater.ref.4.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
blower.ref.1	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.ref.2	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.ref.3	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.ref.4	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.blower.ref.1	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF OPERATIONAL FLT LOW PRESSURE CLOGGED BLOCKED SAND CYCLE PRESSURE SENSOR FLT)
state.blower.ref.2	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF OPERATIONAL FLT LOW PRESSURE CLOGGED BLOCKED SAND CYCLE PRESSURE SENSOR FLT)
state.blower.ref.3	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF OPERATIONAL FLT LOW PRESSURE CLOGGED BLOCKED SAND CYCLE PRESSURE SENSOR FLT)
state.blower.ref.4	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF OPERATIONAL FLT LOW PRESSURE CLOGGED BLOCKED SAND CYCLE PRESSURE SENSOR FLT)
blower.ref.1.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
blower.ref.2.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
blower.ref.3.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
blower.ref.4.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temp.ref.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.ref.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
temp.ref.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.ref.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
delta.ref.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
delta.ref.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
delta.ref.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
delta.ref.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dew	BOOLEAN	R/O	BooleanRange R/O (true false)
state.dew	CHOICE		EnumRange (UNINITIALIZED DISABLED OFF ACTIVE)
internal.dp7	TEXT	R/O	StringRange R/O
internal.dp9	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Rain Sensor)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Feed Blower)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Feed Heater)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Main Reflector Heater)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Reflector 1 Heater)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reflector 2 Heater)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reflector 3 Heater)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Reflector 4 Heater)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Reflector 1 Blower)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Reflector 2 Blower)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Reflector 3 Blower)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Reflector 4 Blower)

9.16 ASCSignal-NGC-Hub-AC

This page describes the device driver and the device window for the hub air condition system integrated in ASC Signal NGC 4000 ACU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements as well as the climate control units state
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given

delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

- highTempAlarm --- Defines the ambient temperatures high alarm limit.
- lowTempAlarm --- Defines the ambient temperatures low alarm limit.
- fan.setpoint --- Defines the temperature setpoint in '°C' for activating the fan.
- fan.diffSetpoint --- Defines the differential temperature setpoint in '°C' for how long the fan stays on.
- cool.setpoint --- Defines the temperature setpoint in '°C' for activating the a/c cooling.
- cool.diffSetpoint.1 --- Defines the differential temperature setpoint in '°C' for how long a/c cooling stays on.
- cool.diffSetpoint.2 --- Defines the differential temperature setpoint in °C for how long a/c cooling stays on
- heat.setpoint --- Defines the temperature setpoint in '°C' for activating the heating furnace
- heat.diffSetpoint.1 --- Defines the heating differential temperature setpoint in '°C' for how long heating furnace stays on.
- heat.diffSetpoint.2 --- Defines the heating differential temperature setpoint in '°C' for how long heating furnace stays on.
- loPressure.setpoint --- Defines the compressor low pressure alarm setpoint.
- hiPressure.setpoint --- Defines the compressor high pressure alarm setpoint.
- humidity.setpoint --- Defines the humidity cycle differential setpoint for how long furnace or cooling stay on.
- humidity.diffSetpoint --- Defines the humidity cycle setpoint
- unitA.voltage --- Defines the output voltage unit B.
- unitB.voltage --- Defines the output voltage unit B.
- switchoverTime --- Defines the auto switchover interval in fixed steps.
- compressorDelay --- Defines the compressor delay time in fixed steps.
- fanMode --- Defines the hub a/c fan mode.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.highTempAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.lowTempAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.fan.setpoint	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.fan.diffSetpoint	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.cool.setpoint	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.cool.diffSetpoint.1	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.cool.diffSetpoint.2	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.heat.setpoint	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.heat.diffSetpoint.1	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.heat.diffSetpoint.2	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.loPressure.setpoint	INTEGER	SETUP	IntegerRange (0 .. 0)
config.hiPressure.setpoint	INTEGER	SETUP	IntegerRange (0 .. 0)
config.humidity.setpoint	INTEGER	SETUP	IntegerRange (0 .. 0)
config.humidity.diffSetpoint	INTEGER	SETUP	IntegerRange (0 .. 0)
config.unitA.voltage	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.unitB.voltage	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.switchoverTime	CHOICE	SETUP	EnumRange (8hrs 24hrs 7dys 30dys)
config.compressorDelay	CHOICE	SETUP	EnumRange (0sec 30sec 60sec 90sec 120sec 150sec 180sec 210sec 240sec 270sec 300sec)
config.fanMode	CHOICE	SETUP	EnumRange (AUTO FAN ON)
temp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan	BOOLEAN	R/O	BooleanRange R/O (true false)
state.fan	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF BLOCKED)
fan.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ac	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ac	CHOICE	R/O	EnumRange R/O (UNINITIALIZED DISABLED OFF ACTIVE COAST ON OVERIDE ON TEMP SENSOR FLT COMMUNICATION FORCED ON FORCED OFF BLOCKED AUTO ON HEAT ON COLD ON FAN ONLY)
ac.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
pressure.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cool.1	BOOLEAN	R/O	BooleanRange R/O (true false)
cool.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heat.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heat.2	BOOLEAN	R/O	BooleanRange R/O (true false)
fan.1	BOOLEAN	R/O	BooleanRange R/O (true false)
fan.2	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
reheat.1	BOOLEAN	R/O	BooleanRange R/O (true false)
reheat.2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Compressor A)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Compressor B)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Heater A)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Heater B)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reheat A)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Reheat B)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low pressure A)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Low pressure B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High pressure A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (High pressure B)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Pressure Transducer A)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Pressure Transducer B)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Pressure Transducer A FLT)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Pressure Transducer B FLT)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Temperature Sensor)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (Humidity Sensor)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Low Hub Temp.)
faults.20	ALARM	R/O	AlarmFlagRange R/O (High Hub Temp.)
faults.21	ALARM	R/O	AlarmFlagRange R/O (HubAC Comms)
faults.22	ALARM	R/O	AlarmFlagRange R/O (BMS Latched)

9.17 ASCSignal-NGC-RedundancyModule

This page describes an addon device driver and the device window for the ASC Signal NGC ACU redundancy module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the redundancy switching parameters
- --- Contains the LNA/LNB meter readings.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

- *parameter* --- *description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If

both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.overallConfig	TEXT	R/O	StringRange R/O
info.block1Config	TEXT	R/O	StringRange R/O
info.block2Config	TEXT	R/O	StringRange R/O
info.block1Name	TEXT	R/O	StringRange R/O
info.block2Name	TEXT	R/O	StringRange R/O
info.block1Type	TEXT	R/O	StringRange R/O
info.block2Type	TEXT	R/O	StringRange R/O
internal.pos1	CHOICE	R/O	EnumRange R/O (F A B)
internal.pos2	CHOICE	R/O	EnumRange R/O (F A B)
internal.flt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.bus1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.bus2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state	CHOICE		EnumRange (ABSENT DISABLED FAULT OPERATIONAL)
reset	TEXT	NOPRESET	StringRange
current.margin	INTEGER		IntegerRange (0 .. 2000)
nom.current.1	INTEGER		IntegerRange (0 .. 2000)
nom.current.2	INTEGER		IntegerRange (0 .. 2000)
nom.current.3	INTEGER		IntegerRange (0 .. 2000)
nom.current.4	INTEGER		IntegerRange (0 .. 2000)
protection.1	CHOICE		EnumRange (MANUAL AUTO)
protection.2	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Main power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Red. power supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch 1)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Protection block 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protection block 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Redundancy Module)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fuse)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fuse)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fuse)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fuse)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Low Current)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Low Current)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Low Current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Low Current)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Unit 1 High Current)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Unit 2 High Current)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Unit 3 High Current)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (Unit 4 High Current)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Unit 1 External)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Unit 2 External)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Unit 3 External)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Unit 4 External)
faults.31	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB1 Latched)
faults.32	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB2 Latched)
faults.33	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB3 Latched)
faults.34	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB4 Latched)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fuse Latched)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fuse Latched)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fuse Latched)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fuse Latched)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Low Current Latched)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Low Current Latched)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Low Current Latched)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Low Current Latched)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Unit 1 High Current Latched)

name	*type*	*flags*	*range*
faults.44	ALARM	R/O	AlarmFlagRange R/O (Unit 2 High Current Latched)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Unit 3 High Current Latched)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Unit 4 High Current Latched)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Unit 1 External Latched)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Unit 2 External Latched)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Unit 3 External Latched)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Unit 4 External Latched)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Redundancy 1-1)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Redundancy 1-2)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Redundancy 1-3)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Redundancy 1-4)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Redundancy 2-1)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Redundancy 2-2)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Redundancy 2-3)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Redundancy 2-4)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Redundancy 3-1)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Redundancy 3-2)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Redundancy 3-3)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Redundancy 3-4)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Redundancy 4-1)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Redundancy 4-2)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Redundancy 4-3)

name	*type*	*flags*	*range*
faults.66	ALARM	R/O	AlarmFlagRange R/O (Redundancy 4-4)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Switch 1-1)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Switch 1-2)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Switch 2-1)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Switch 2-2)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Switch 3-1)

9.18 ASCSignal-NGC-Temperature

This page describes the device driver and the device window for up to 16 temperature sensors integrated in ASC SignalNGC 4000 ACU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all sensor measurements as well as their thresholds
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
- sens.xx.hiThreshold --- Defines the high temperature threshold in '°C' for the respective sensor.
- sens.xx.loThreshold --- Defines the low temperature threshold in '°C' for the respective sensor.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sens.01.serial	TEXT	R/O	StringRange R/O
info.sens.02.serial	TEXT	R/O	StringRange R/O
info.sens.03.serial	TEXT	R/O	StringRange R/O
info.sens.04.serial	TEXT	R/O	StringRange R/O
info.sens.05.serial	TEXT	R/O	StringRange R/O
info.sens.06.serial	TEXT	R/O	StringRange R/O
info.sens.07.serial	TEXT	R/O	StringRange R/O
info.sens.08.serial	TEXT	R/O	StringRange R/O
info.sens.09.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.sens.10.serial	TEXT	R/O	StringRange R/O
info.sens.11.serial	TEXT	R/O	StringRange R/O
info.sens.12.serial	TEXT	R/O	StringRange R/O
info.sens.13.serial	TEXT	R/O	StringRange R/O
info.sens.14.serial	TEXT	R/O	StringRange R/O
info.sens.15.serial	TEXT	R/O	StringRange R/O
info.sens.16.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.sens.01.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.01.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.02.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.02.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.03.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.03.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.04.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.04.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.05.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.05.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.06.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.06.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.07.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.07.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.08.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.08.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.09.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.09.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)

name	*type*	*flags*	*range*
config.sens.10.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.10.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.11.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.11.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.12.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.12.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.13.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.13.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.14.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.14.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.15.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.15.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.16.hiThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
config.sens.16.loThreshold	FLOAT	SETUP	DoubleRange (-55.0 .. 125.0)
sens.01.name	TEXT		StringRange
sens.02.name	TEXT		StringRange
sens.03.name	TEXT		StringRange
sens.04.name	TEXT		StringRange
sens.05.name	TEXT		StringRange
sens.06.name	TEXT		StringRange
sens.07.name	TEXT		StringRange
sens.08.name	TEXT		StringRange
sens.09.name	TEXT		StringRange
sens.10.name	TEXT		StringRange
sens.11.name	TEXT		StringRange
sens.12.name	TEXT		StringRange
sens.13.name	TEXT		StringRange

name	*type*	*flags*	*range*
sens.14.name	TEXT		StringRange
sens.15.name	TEXT		StringRange
sens.16.name	TEXT		StringRange
sens.01.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.02.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.03.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.04.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.05.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.06.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.07.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.08.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.09.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.10.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.11.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.12.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.13.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.14.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.15.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sens.16.meas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 Low Temp.)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 High Temp.)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 Low Temp.)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 High Temp.)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 Low Temp.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 High Temp.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 Low Temp.)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 High Temp.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 Low Temp.)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 High Temp.)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 Low Temp.)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 High Temp.)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 Low Temp.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 High Temp.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 Low Temp.)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 High Temp.)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 Low Temp.)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 High Temp.)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 Low Temp.)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 High Temp.)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 Low Temp.)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 High Temp.)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 Low Temp.)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 High Temp.)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 Low Temp.)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 High Temp.)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 Low Temp.)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 High Temp.)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 Low Temp.)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 High Temp.)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 Low Temp.)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 High Temp.)

9.19 ASCSignal-NGC-Weather

This page describes the read only device driver and the device window for the ASC Signal NGC 4000 ACU integrated weather station.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements of the weather station

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- wind.hiThreshold --- Defines the high wind threshold in m/s. If the measured value exceeds the limit a high wind fault is raised.
- temp.loThreshold --- Defines the low temperature threshold in '°C'. If the measured value fall below the limit a low temperature fault is raised.
- temp.hiThreshold --- Defines the high temperature threshold in '°C'. If the measured value exceeds the limit a high temperature fault is raised.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.wind.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 60.0)
config.temp.loThreshold	FLOAT	SAVE SETUP	DoubleRange (-52.0 .. 60.0)
config.temp.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (-52.0 .. 60.0)
speed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
direction	INTEGER	R/O	IntegerRange R/O (0 .. 360)
windRose	TEXT	R/O	StringRange R/O
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
solarFlux	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (High Wind Speed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low Temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High Temperature)

9.20 Acorde-ACRS-IDU

This page describes the device driver and the device window for the Acorde LNA/LNB Redundancy System Controller via SNMP.

Tested with the following 2:1 redundancy model:

- ACRS-IDU-V1-MNU-04-00
- FW ACFIRM-M9-IDU-V218-COD-05-00

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the redundancy settings.
- --- The second page shows the voltage and current measurement values.
- --- The third page shows the voltage and current threshold values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
model	Defines the model redundancy type: 1:1 or 2:1
remote	Sets the unit to remote or local access
hasTrackingLna	Defines if the unit is equipped with a separate tracking LNA. Set to NO by Default.
remote	Enable/Disable latched faults

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.alias	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.model	CHOICE	SETUP	EnumRange (1:1 1:2)
config.remote	CHOICE	SETUP	EnumRange (REMOTE LOCAL)

name	*type*	*flags*	*range*
config.hasTrackingLna	CHOICE	SAVE SETUP	EnumRange (NO YES)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltagePsu.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltagePsu.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.1.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.2.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.3.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.4.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.1.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.2.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.3.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
voltage.4.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.1.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.2.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.3.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.4.lowThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.1.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.2.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
current.3.highThd	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
current.4.highThd	FLOAT		DoubleRange (0.0 .. 0.0)
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (2-STBY 1-STBY 3-STBY A B)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
priority	CHOICE		EnumRange (NONE UNIT1 UNIT2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (User Memory)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal memory)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Voltage)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Voltage)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Voltage)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Voltage)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Current)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Current)

name	*type*	*flags*	*range*
faults.33	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Current)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Current)

9.21 Acorde-BUC-ACCVT

This page describes the device driver and the device window for the for the Acorde (dual-) block upconverter via SNMP.

It was developed based on `ACCVT-LKa-Ex-Vx-MIB-01-06.mib`

- Tested with model: `ACCVT-LKa-E28-V4`
- Firmware Version: `ACFIRM-M9-CONV-V282-COD-03-20`

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The operational parameters
- --- The meter readings. Temperature/Voltage/Current.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**
address	The SNMP address specification for the device (details below).
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extRef	Allows to switch unit between internal or external reference source, automatic mode and automatic fallback mode.
gainRange	Allows to adjust the gain range of the respective unit. The default range is 30dB.

parameter	description
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The `\n` character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write communit name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.alias	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO AUTO-FALLBACK)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain.1	FLOAT		DoubleRange (0.00 .. 30.00)
tx.gain.2	FLOAT		DoubleRange (0.00 .. 30.00)
tx.attn.1	FLOAT		DoubleRange (0.00 .. 30.00)
tx.attn.2	FLOAT		DoubleRange (0.00 .. 30.00)
tx.eq.1	FLOAT		DoubleRange (0.0 .. 9.0)
tx.eq.2	FLOAT		DoubleRange (0.0 .. 9.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
reset	TEXT	NOPRESET	StringRange
recover	TEXT	NOPRESET	StringRange DIS
internal.gain1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gain2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.old	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (User Memory)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Factory Memory)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Init Voltages)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low Band Lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (High Band Lock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (OCXO Lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Int Communication)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Overcurrent)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Undercurrent)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Overvoltage)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Undervoltage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temperature Warn)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Int Memory Warn)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Reference Warn)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.22 Actia-Sodielec-DNT-HPA

This page describes the device driver and the device window for the Actia Sodielec DNT TWT amplifiers. The device driver actually supports the following models:

- Actia Sodielec DNT503C

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address must be specified as a decimal number in the range 32 .. 127. The device address entered here is the address (0 ..95) set at the device plus 32.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *19200,E,7,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
tx.attn	FLOAT	NOCOMPARE	DoubleRange (0.00 .. 64.00)
tx.gain	FLOAT	NOCOMPARE	DoubleRange (0.00 .. 64.00)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.stat.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stat.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stat.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stat.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stat.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.stat.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stat.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.off	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.stby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.preh	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.rfon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.attn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Heater max V/I)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Heater min V/I)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pulse TOut)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Stby timeout)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Blower)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HSTS)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Input power)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output power)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Refl. power)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Body current)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Beam control)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Collector)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Helix)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Heater)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Vck max)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Vck min)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Vak max)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Vak min)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Vhk max)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Vhk min)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Saving)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.27	ALARM	R/O	AlarmFlagRange R/O (System)
faults.28	ALARM	R/O	AlarmFlagRange R/O (WG arc test)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Ih / body test)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Panels)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Remote control)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.23 Actia-Sodielec-DSS1300

This page describes the device driver and the device window for the*Actia Sodielec DSS1300* switch controller.

The *Actia-Sodielec-DSS1300* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Actia-Sodielec-DSS1300* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Actia-Sodielec-DSS1300* device driver as a low level driver to access the physical I/O ports.

Input ports

The variable 'inputs' refers to the input ports 1..16 monitored by the device. The least significant bit reports the state of input 1, the 2^15 bit reports the state of input 16.

Output ports

The variable 'outputs' is used to control the outputs of the device. The least significant bit in 'outputs' corresponds to output 1, the most significant bit to output 16.

- --- **The DSS1300 switch controller does not provide any means to read back the actual state of an output. For this reason all outputs read "0" / "inactive" after the start of the software.**

Individual output variables

Beside of using the 'outputs' variable together with the sat-nms framework for M&C of plain digital I/O functions, each output may controlled directly through an individual parameter. The variables *out01* .. *out16* accept an integer value in the range 0 .. 255 and directly perform the device's output command. The value passed to the variable select the way how to actuate the output:

value	***description***
0	switches the output OFF
1..254	generates a pulse of a defined length. The output is switched ON for*value * 10msecs* , then switched OFF again.
255	switches the output permanently ON

The individual output variables may be used to resemble the device's front panel buttons.

- --- **An application must not use the 'outputs' variable and the individual outputs at the same time. Setting individual outputs will mess up the**

behaviour of the 'outputs' variable and vice versa.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	The device address must be specified as a decimal number in the range 32 .. 127. The device address entered here is the address (0 ..95) set at the device plus 32. Factory default of the device is 126 (94 at the device).
serial ifc	The serial interface settings must match the line parameters programmed in the device. Factory default of the device is 19200,e,7,1

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.revision	TEXT	R/O	StringRange R/O
initialized	TEXT	R/O	StringRange R/O
inputs	HEX	R/O	HexRange R/O (0 .. 0)
outputs	HEX		HexRange (0 .. 0)
out00	INTEGER		IntegerRange (0 .. 255)
out01	INTEGER		IntegerRange (0 .. 255)
out02	INTEGER		IntegerRange (0 .. 255)
out03	INTEGER		IntegerRange (0 .. 255)
out04	INTEGER		IntegerRange (0 .. 255)
out05	INTEGER		IntegerRange (0 .. 255)
out06	INTEGER		IntegerRange (0 .. 255)
out07	INTEGER		IntegerRange (0 .. 255)
out08	INTEGER		IntegerRange (0 .. 255)
out09	INTEGER		IntegerRange (0 .. 255)
out0A	INTEGER		IntegerRange (0 .. 255)
out0B	INTEGER		IntegerRange (0 .. 255)
out0C	INTEGER		IntegerRange (0 .. 255)
out0D	INTEGER		IntegerRange (0 .. 255)
out0E	INTEGER		IntegerRange (0 .. 255)
out0F	INTEGER		IntegerRange (0 .. 255)

9.24 Adtec-Afiniti-Decoder

This page describes the device driver and the device window for the Adtec Afiniti Decoder device via SNMP. It supports one decoding instance. For the second decoder instance another device is necessary.

Additionally it is necessary to select the slot where decoding card is installed via device setup config.slot. The dedicated decoder can be selected via config.channel.

For 4k applications that use a single Video PID the Decoder must be set to Single Mode

(channel.mode = SINGLE). In single channel mode the 4 SDI outputs will be independent and not mirrored pairs.

The unit does not support BISS Decryption.

It is necessary to preconfigure the Adtec Afiniti Decoder via the webpage:

- ASI-Inputs 'TS Mux Rate' to a suitable value
- ASI-Inputs 'Share Services with All Outputs' to 'ON'

The driver was tested with the following sw releases and it is not backwards compatible to earlier releases.

- 4.4.6

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The main decoding parameters, including service selection.
- --- The video parameters
- --- The audio parameters
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the decoder.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
    - The read/write-community strings factory defaults are 'public' for the read community, 'adtecuser' for the write community. Hence a typical address

specification for a Adtec-Afiniti may be *10.0.0.1 public adtecuser*.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- slot --- Defines the slot where encoding card is installed 1-6.
- channel --- Defines the used encoder for this device from the 1-4.
- output --- Defines the index of the ts output which should be allocated to the chosen decoding channel.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.mainModel	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mainVersion	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.mainSerial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7)
config.channel	CHOICE	SAVE SETUP	EnumRange (1 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.channel	TEXT	R/O	StringRange R/O
internal.output	TEXT	R/O	StringRange R/O
internal.channelMode	CHOICE	R/O	EnumRange R/O (SINGLE MULTI)
internal.inputId	INTEGER		IntegerRange (0 .. 0)
channel.mode	CHOICE		EnumRange (SINGLE MULTI)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
programNo	INTEGER		IntegerRange (-1 .. 65535)
programList	CHOICE		EnumRange (NO SERVICE)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER		IntegerRange (0 .. 0)
video.uhdMode	CHOICE		EnumRange (OFF 2-SI SQUARE)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.profile	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
video.format	TEXT	R/O	StringRange R/O
video.errors	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
video.demuxErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.3.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.4.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.5.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.6.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.7.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.bitrate	INTEGER		IntegerRange (0 .. 0)
audio.8.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.bitrate	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)

9.25 Adtec-Afinity-Encoder

This page describes the device driver and the device window for the Adtec Afiniti Encoder device via SNMP. It supports RTP/UDP and SRT output.

When using UHD (channel.mode = SINGLE) encoders instances 2-4 are readonly.

Driver support one HD encoding instance. For each additional instance another device is needed. Additionally it is necessary to select the slot where encoding card is installed via device setup config.slot. The dedicated Output can be selected via config.output

The driver was tested with the following sw releases:

- 4.2.0 It is not backwards compatible to earlier releases

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the MPEG output parameter settings and the operation control.
- --- Contains RTP/UDP and SRT output parameters and statistics.
- --- Contains the conditional access (BISS) parameters.
- --- Contains video in/output parameters from the MPEG stream.

- --- Contains the parameters concerning the audio channel 1 encoding.
- --- Contains the parameters concerning the audio channel 2 encoding.
- --- Contains the parameters concerning the audio channel 3 encoding.
- --- Contains the parameters concerning the audio channel 4 encoding.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- slot --- Defines the slot where encoding card is installed 1-6.
- channel --- Defines the used encoder for this device from the 1-4.
- output --- Defines the index of the ts output which should be allocated to the chosen encoding channel.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.mainModel	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mainVersion	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.mainSerial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.channel	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
config.outputType	CHOICE	SAVE SETUP	EnumRange (ASI IP)
config.output	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.channel	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.output	TEXT	R/O	StringRange R/O
internal.channelMode	CHOICE	R/O	EnumRange R/O (SINGLE MULTI)
internal.bissmode	TEXT	R/O	StringRange R/O
internal.encryption	CHOICE	R/O	EnumRange R/O (ON OFF)
internal.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hSize	TEXT	R/O	StringRange R/O
internal.vSize	TEXT	R/O	StringRange R/O
internal.frame	TEXT	R/O	StringRange R/O
internal.prog	TEXT	R/O	StringRange R/O
internal.audio1	TEXT	R/O	StringRange R/O
internal.audio2	TEXT	R/O	StringRange R/O
internal.audio3	TEXT	R/O	StringRange R/O
internal.audio4	TEXT	R/O	StringRange R/O
internal.protocol	TEXT	R/O	StringRange R/O
video.codec	CHOICE		EnumRange (MPEG2 MPEG4-AVC HEVC)
channel.mode	CHOICE		EnumRange (SINGLE MULTI)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.source	TEXT	R/O	StringRange R/O
output.latency	CHOICE		EnumRange (NORMAL LOW VERY-LOW LONG)
output.latencyTrim	INTEGER		IntegerRange (-400 .. 100)
output.datarate	FLOAT		DoubleRange (0.000 .. 0.000)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.programName	TEXT		StringRange
output.service	TEXT		StringRange
output.provider	TEXT		StringRange

name	*type*	*flags*	*range*
output.tsid	INTEGER		IntegerRange (0 .. 65535)
output.nitpid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (0 .. 8191)
output.muxrate	FLOAT	R/O	DoubleRange R/O (1.000 .. 210.000)
video.pid	INTEGER		IntegerRange (0 .. 8190)
video.startup	CHOICE		EnumRange (ON OFF)
video.chroma	CHOICE		EnumRange (4:2:0 4:2:2)
video.bitdepth	CHOICE		EnumRange (8-Bit 10-Bit)
video.aspectRatio	CHOICE		EnumRange (4:3 16:9)
video.gopType	CHOICE		EnumRange (CLOSED OPEN)
video.gopStruct	CHOICE		EnumRange (IP IBP IBBP)
video.gopLength	INTEGER		IntegerRange (1 .. 64)
video.autoFill	CHOICE		EnumRange (OFF ON)
video.loss	CHOICE		EnumRange (STOP ENCODING BARS BARS+ID)
video.lossFormat	CHOICE		EnumRange (480i59.94 576i50 720p59.94 720p50 1080i59.94 1080i50 1080p59.94 1080p50)
video.datarate	FLOAT		DoubleRange (1.000 .. 94.916)
video.uhdMode	CHOICE		EnumRange (OFF 2-SI SQUARE)
video.uhdlossFormat	CHOICE		EnumRange (2160p59.94 2160p50 2160p29.97 2160p25)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.injectedid	TEXT	NOPRESET	StringRange
ipout.1.enable	CHOICE		EnumRange (ON OFF)
ipout.1.interface	CHOICE		EnumRange (PORT-1 PORT-2)

name	*type*	*flags*	*range*
ipout.1.datarate	FLOAT		DoubleRange (1.000 .. 210.000)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.protocol	CHOICE		EnumRange (RTP UDP SRT)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.buffer	INTEGER		IntegerRange (0 .. 500)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.latency	INTEGER		IntegerRange (0 .. 60000)
srtout.1.packetDelivery	CHOICE		EnumRange (OFF ON DEFAULT)
srtout.1.peerLatency	INTEGER		IntegerRange (0 .. 60000)
srtout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
srtout.1.key	TEXT	NOPRESET	StringRange
srtout.1.state	TEXT	R/O	StringRange R/O
srtout.1.rtt	TEXT	R/O	StringRange R/O
srtout.1.datarate	TEXT	R/O	StringRange R/O
srtout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.buffer	TEXT	R/O	StringRange R/O
audio.1.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.delay	INTEGER		IntegerRange (-800 .. 800)

name	*type*	*flags*	*range*
audio.1.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.1.profile	CHOICE		EnumRange (MPEG2-L2 03 MPEG2-L2 04 DD-AC3-81 DD-AC3-06 DOLBY-E AAC-LC AAC-HE AAC-HEv2 LPCM)
audio.1.language	TEXT		StringRange
audio.1.level	INTEGER		IntegerRange (-18 .. 8)
audio.1.aacPhase	CHOICE		EnumRange (YES NO)
audio.1.aacFormat	CHOICE		EnumRange (ADTS-MP2 ADTS-MP4 LOAS)
audio.1.aacChannel	CHOICE		EnumRange (STEREO)
audio.1.aacSpeech	CHOICE		EnumRange (YES NO)
audio.1.aacStereo	CHOICE		EnumRange (YES NO)
audio.1.aacCrc	CHOICE		EnumRange (YES NO)
audio.1.dolbyCoding	CHOICE		EnumRange (1/0 2/0)
audio.1.dolbyBitstream	CHOICE		EnumRange (MAIN MAIN DIALOGUE VISUALLY IMPAIRED HEARING IMPAIRED DIALOGUE COMMENTARY EMERGENCY FLASH VOICE OVER MAIN KARAOKE)
audio.1.dolbyNormalization	INTEGER		IntegerRange (1 .. 31)
audio.1.dolbyProduction	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.1.dolbyLevel	INTEGER		IntegerRange (1 .. 31)
audio.1.dolbyRoom	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)

name	*type*	*flags*	*range*
audio.1.dolbyCopyright	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.1.dolbyOriginal	CHOICE		EnumRange (COPY ORIGINAL)
audio.1.dolbyLineMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.1.dolbyRFMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.1.dolbyModulation	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.1.dolbyLpf	CHOICE		EnumRange (OFF ON)
audio.1.dolbyDCFilter	CHOICE		EnumRange (OFF ON)
audio.2.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.delay	INTEGER		IntegerRange (-800 .. 800)
audio.2.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.2.profile	CHOICE		EnumRange (MPEG2-L2 03 MPEG2-L2 04 DD-AC3-81 DD-AC3-06 DOLBY-E AAC-LC AAC-HE AAC-HEv2 LPCM)
audio.2.language	TEXT		StringRange
audio.2.level	INTEGER		IntegerRange (-18 .. 8)
audio.2.aacPhase	CHOICE		EnumRange (YES NO)

name	*type*	*flags*	*range*
audio.2.aacFormat	CHOICE		EnumRange (ADTS-MP2 ADTS-MP4 LOAS)
audio.2.aacChannel	CHOICE		EnumRange (STEREO)
audio.2.aacSpeech	CHOICE		EnumRange (YES NO)
audio.2.aacStereo	CHOICE		EnumRange (YES NO)
audio.2.aacCrc	CHOICE		EnumRange (YES NO)
audio.2.dolbyCoding	CHOICE		EnumRange (1/0 2/0)
audio.2.dolbyBitstream	CHOICE		EnumRange (MAIN MAIN DIALOGUE VISUALLY IMPAIRED HEARING IMPAIRED DIALOGUE COMMENTARY EMERGENCY FLASH VOICE OVER MAIN KARAOKE)
audio.2.dolbyNormalization	INTEGER		IntegerRange (1 .. 31)
audio.2.dolbyProduction	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.2.dolbyLevel	INTEGER		IntegerRange (1 .. 31)
audio.2.dolbyRoom	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.2.dolbyCopyright	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.2.dolbyOriginal	CHOICE		EnumRange (COPY ORIGINAL)
audio.2.dolbyLineMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.2.dolbyRFMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.2.dolbyModulation	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.2.dolbyLpf	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
audio.2.dolbyDCFilter	CHOICE		EnumRange (OFF ON)
audio.3.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.delay	INTEGER		IntegerRange (-800 .. 800)
audio.3.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.3.profile	CHOICE		EnumRange (MPEG2-L2 03 MPEG2-L2 04 DD-AC3-81 DD-AC3-06 DOLBY-E AAC-LC AAC-HE AAC-HEv2 LPCM)
audio.3.language	TEXT		StringRange
audio.3.level	INTEGER		IntegerRange (-18 .. 8)
audio.3.aacPhase	CHOICE		EnumRange (YES NO)
audio.3.aacFormat	CHOICE		EnumRange (ADTS-MP2 ADTS-MP4 LOAS)
audio.3.aacChannel	CHOICE		EnumRange (STEREO)
audio.3.aacSpeech	CHOICE		EnumRange (YES NO)
audio.3.aacStereo	CHOICE		EnumRange (YES NO)
audio.3.aacCrc	CHOICE		EnumRange (YES NO)
audio.3.dolbyCoding	CHOICE		EnumRange (1/0 2/0)
audio.3.dolbyBitstream	CHOICE		EnumRange (MAIN MAIN DIALOGUE VISUALLY IMPAIRED HEARING IMPAIRED DIALOGUE COMMENTARY EMERGENCY FLASH VOICE OVER MAIN KARAOKE)
audio.3.dolbyNormalization	INTEGER		IntegerRange (1 .. 31)

name	*type*	*flags*	*range*
audio.3.dolbyProduction	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.3.dolbyLevel	INTEGER		IntegerRange (1 .. 31)
audio.3.dolbyRoom	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.3.dolbyCopyright	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.3.dolbyOriginal	CHOICE		EnumRange (COPY ORIGINAL)
audio.3.dolbyLineMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.3.dolbyRFMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.3.dolbyModulation	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.3.dolbyLpf	CHOICE		EnumRange (OFF ON)
audio.3.dolbyDCFilter	CHOICE		EnumRange (OFF ON)
audio.4.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.delay	INTEGER		IntegerRange (-800 .. 800)
audio.4.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)

name	*type*	*flags*	*range*
audio.4.profile	CHOICE		EnumRange (MPEG2-L2 03 MPEG2-L2 04 DD-AC3-81 DD-AC3-06 DOLBY-E AAC-LC AAC-HE AAC-HEv2 LPCM)
audio.4.language	TEXT		StringRange
audio.4.level	INTEGER		IntegerRange (-18 .. 8)
audio.4.aacPhase	CHOICE		EnumRange (YES NO)
audio.4.aacFormat	CHOICE		EnumRange (ADTS-MP2 ADTS-MP4 LOAS)
audio.4.aacChannel	CHOICE		EnumRange (STEREO)
audio.4.aacSpeech	CHOICE		EnumRange (YES NO)
audio.4.aacStereo	CHOICE		EnumRange (YES NO)
audio.4.aacCrc	CHOICE		EnumRange (YES NO)
audio.4.dolbyCoding	CHOICE		EnumRange (1/0 2/0)
audio.4.dolbyBitstream	CHOICE		EnumRange (MAIN MAIN DIALOGUE VISUALLY IMPAIRED HEARING IMPAIRED DIALOGUE COMMENTARY EMERGENCY FLASH VOICE OVER MAIN KARAOKE)
audio.4.dolbyNormalization	INTEGER		IntegerRange (1 .. 31)
audio.4.dolbyProduction	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.4.dolbyLevel	INTEGER		IntegerRange (1 .. 31)
audio.4.dolbyRoom	CHOICE		EnumRange (NOT INDICATED LARGE ROOM SMALL ROOM)
audio.4.dolbyCopyright	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.4.dolbyOriginal	CHOICE		EnumRange (COPY ORIGINAL)
audio.4.dolbyLineMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)

name	*type*	*flags*	*range*
audio.4.dolbyRFMode	CHOICE		EnumRange (COPY FILM STANDARD FILM LIGHT MUSIC STANDARD MUSIC LIGHT SPEECH)
audio.4.dolbyModulation	CHOICE		EnumRange (NO PROTECTION PROTECTION)
audio.4.dolbyLpf	CHOICE		EnumRange (OFF ON)
audio.4.dolbyDCFilter	CHOICE		EnumRange (OFF ON)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)

9.26 Adtec-EN81

This page describes the device driver and the device window for the Adtec EN81 video encoder family.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters (only present for devices equipped with a modulator unit)
- --- MPEG output parameters
- --- C/A encryption parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the Adtec-EN81 front panel or the web interface of the device. Factory defaults are 'adtec' for the read community, 'none' for the write community. Hence a typical address specification for a Adtec-EN81 may be *10.0.0.1 adtec none* .

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- withModulator --- If the driver controls a unit equipped with an L-band modulator, you have to set this to 'YES'. This enables the modulator device screen and the parameters in it. Enabling the modulator parameters for a device without modulator will cause a communication fault.
- loFreq --- Sets the upconverter LO frequency. Please note, that this only sets the LO frequency used by the sat-nms software, the value used by the device's web interface has to be set separately. Setting the LO frequency to zero lets the driver show/set the transmit frequency in terms of L-band frequency.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.product	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.withModulator	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.loFreq	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loPol	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERT)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.type	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
output.datarate	FLOAT		DoubleRange (1.00000 .. 80.00000)
output.asiMode	CHOICE		EnumRange (CONTINOUS ENCODE ONLY)
output.latency	CHOICE		EnumRange (LONG NORMAL LOW VERY LOW)
output.latencyTrim	INTEGER		IntegerRange (-400 .. 200)
output.tables	CHOICE		EnumRange (DVB MPEG ATSC)

name	*type*	*flags*	*range*
output.program	INTEGER		IntegerRange (1 .. 65535)
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)
mod.frequency	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
mod.rfon	CHOICE		EnumRange DIS (OFF ON)
mod.power	FLOAT		DoubleRange DIS (-35.0 .. 5.0)
mod.standard	CHOICE		EnumRange DIS (DVB-S DVB-S2)
mod.modulation	CHOICE		EnumRange DIS (C/W ON CLOCK/8 CLOCK/4 CLOCK/16)
mod.fec	CHOICE		EnumRange DIS (QPSK-NONE QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10)
mod.fec2	CHOICE		EnumRange DIS (1/2 1/3 1/4 2/3 2/5 3/4 3/5 4/5 5/6 6/7 7/8 8/9 9/10 NONE)
mod.modulation2	CHOICE		EnumRange DIS (QPSK 16APSK 32APSK 16QAM 8PSK)
mod.symbolrate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
mod.datarate	FLOAT	R/O	DoubleRange R/O DIS (0.0000 .. 0.0000)
mod.bandwidth	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
mod.spectrum	CHOICE		EnumRange DIS (NORM INVERT)

name	*type*	*flags*	*range*
mod.rollOff	CHOICE		EnumRange DIS (35% 25% 20% 15% 10% 5%)
mod.pilot	CHOICE		EnumRange DIS (OFF ON)
mod.frameSize	CHOICE		EnumRange DIS (NORMAL SHORT)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E-UID1 BISS-E-UID2)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.injectedid	TEXT	NOPRESET	StringRange
video.source	CHOICE		EnumRange (SDI COMPOSITE)
video.sdiMode	CHOICE		EnumRange (AUTO SD HD 1.4G)
video.codec	CHOICE		EnumRange (MPEG2 MPEG4 (H264))
video.chroma	CHOICE		EnumRange (4.2.0 4.2.2)
video.deblock	CHOICE		EnumRange (OFF ON)
video.fieldEncoding	CHOICE		EnumRange (FIELD FRAME ADAPTIVE)
video.autoFill	CHOICE		EnumRange (OFF ON)
video.datarate	INTEGER		IntegerRange (0 .. 0)
video.dummyrate	TEXT		StringRange
video.aspect	CHOICE		EnumRange (4:3 16:9 WSS)
video.gopStruct	CHOICE		EnumRange (IBBP IBP IP I)
video.gopLength	INTEGER		IntegerRange (1 .. 30)
video.gopType	CHOICE		EnumRange (OPEN CLOSED)
video.loss	CHOICE		EnumRange (STOP ENCODING BARS BARS+TONES BARS+ID BARS+TONES+ID)
video.pid	INTEGER		IntegerRange (32 .. 8190)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.samplingRate	CHOICE		EnumRange (48000 44100 32000)

name	*type*	*flags*	*range*
audio.1.enable	CHOICE		EnumRange (OFF ENCODE PASSTHRU)
audio.1.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.1.datarate	INTEGER		IntegerRange (64000 .. 640000)
audio.1.lang	TEXT		StringRange
audio.1.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG1 LAYER2 DOLBY E LINEAR-PCM)
audio.1.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.1.feedback	CHOICE		EnumRange (OFF ON GHOST)
audio.1.delay	INTEGER		IntegerRange (-800 .. 800)
audio.1.gain	INTEGER		IntegerRange (-18 .. 7)
audio.1.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 5/6 GROUP 2 - 7/8 GROUP 3 - 9/10 GROUP 3 - 11/12 GROUP 4 - 13/14 GROUP 4 - 15/16)
audio.1.pid	INTEGER		IntegerRange (32 .. 8191)
audio.1.state	CHOICE	R/O	EnumRange R/O (ABSENT UNCONFIGURED IDLING RUNNING)
audio.2.enable	CHOICE		EnumRange (OFF ENCODE PASSTHRU)
audio.2.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.2.datarate	INTEGER		IntegerRange (64000 .. 640000)
audio.2.lang	TEXT		StringRange
audio.2.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG1 LAYER2 DOLBY E LINEAR-PCM)
audio.2.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.2.feedback	CHOICE		EnumRange (OFF ON GHOST)
audio.2.delay	INTEGER		IntegerRange (-800 .. 800)
audio.2.gain	INTEGER		IntegerRange (-18 .. 7)

name	*type*	*flags*	*range*
audio.2.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 5/6 GROUP 2 - 7/8 GROUP 3 - 9/10 GROUP 3 - 11/12 GROUP 4 - 13/14 GROUP 4 - 15/16)
audio.2.pid	INTEGER		IntegerRange (32 .. 8191)
audio.2.state	CHOICE	R/O	EnumRange R/O (ABSENT UNCONFIGURED IDLING RUNNING)
audio.3.enable	CHOICE		EnumRange (OFF ENCODE PASSTHRU)
audio.3.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.3.datarate	INTEGER		IntegerRange (64000 .. 640000)
audio.3.lang	TEXT		StringRange
audio.3.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG1 LAYER2 DOLBY E LINEAR-PCM)
audio.3.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.3.feedback	CHOICE		EnumRange (OFF ON GHOST)
audio.3.delay	INTEGER		IntegerRange (-800 .. 800)
audio.3.gain	INTEGER		IntegerRange (-18 .. 7)
audio.3.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 5/6 GROUP 2 - 7/8 GROUP 3 - 9/10 GROUP 3 - 11/12 GROUP 4 - 13/14 GROUP 4 - 15/16)
audio.3.pid	INTEGER		IntegerRange (32 .. 8191)
audio.3.state	CHOICE	R/O	EnumRange R/O (ABSENT UNCONFIGURED IDLING RUNNING)
audio.4.enable	CHOICE		EnumRange (OFF ENCODE PASSTHRU)
audio.4.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.4.datarate	INTEGER		IntegerRange (64000 .. 640000)
audio.4.lang	TEXT		StringRange

name	*type*	*flags*	*range*
audio.4.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG1 LAYER2 DOLBY E LINEAR-PCM)
audio.4.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.4.feedback	CHOICE		EnumRange (OFF ON GHOST)
audio.4.delay	INTEGER		IntegerRange (-800 .. 800)
audio.4.gain	INTEGER		IntegerRange (-18 .. 7)
audio.4.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 5/6 GROUP 2 - 7/8 GROUP 3 - 9/10 GROUP 3 - 11/12 GROUP 4 - 13/14 GROUP 4 - 15/16)
audio.4.pid	INTEGER		IntegerRange (32 .. 8191)
audio.4.state	CHOICE	R/O	EnumRange R/O (ABSENT UNCONFIGURED IDLING RUNNING)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 4)

9.27 Adtec-RD60

This page describes the device driver and the device window for the Adtec RD60 IRD family.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The main receive parameters, including frequency and service selection.
- --- The conditional access parameters
- --- The video parameters
- --- The audio parameters
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the Adtec-RD60 front panel or the web interface of the device. Factory defaults are 'adtec' for the read community, 'none' for the write community. Hence a typical address specification for a Adtec-EN81 may be *10.0.0.1 adtec none* .

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized to a proved preset value which should not be changed under normal circumstances.
- 22kHz --- Turns the 22 kHz signal in the LNB supply on or off. Only effective if 22kHzAuto is set OFF.
- 22kHzAuto --- Switching this on, makes the driver controlling the 22kHz signal with the receive frequency. For frequencies above the "band edge" 22kHz gets switched on.
- lnbPower --- Switches the LNB power supply ON or OFF.
- lnbVoltage --- Selects the LNB voltage (13V/18V). Only effective if the "lnbPolSwitch" parameter is set OFF.
- lnbPolSwitch --- Selects if the LNB voltage shall be permanently set to the the value selected by "lnbVoltage" (OFF) or if the voltage shall be selected from the "polarization" parameter of the driver.
- loFreqHigh --- The high band LO frequency.
- loFreqLow --- The low band LO frequency.
- bandEdge --- The band edge frequency.
- marginThreshold --- If "state.margin" drops below this value, a "Margin" fault is generated.
- serviceHunt --- Setting this ON lets the receiver use the first available service when receiving a new signal. OFF forces the receiver to get the service with the service ID set.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n"

character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.product	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.serviceHunt	CHOICE	SAVE SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.autoCCM	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.apid	TEXT	R/O	StringRange R/O
internal.ix	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mod	TEXT		StringRange
internal.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input	CHOICE		EnumRange (ASI RF1 RF2 IP)
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK 16QAM AUTO)
fec	CHOICE		EnumRange (NOFEC 1/2 1/3 1/4 2/3 2/5 3/4 3/5 4/5 5/6 6/7 7/8 8/9 9/10 AUTO)
searchRange	INTEGER		IntegerRange (0 .. 0)
polarization	CHOICE		EnumRange (X Y)
rolloff	CHOICE		EnumRange (AUTO 20% 25% 35%)
ccm	CHOICE		EnumRange (AUTO CCM)
pilot	CHOICE		EnumRange (AUTO ON OFF)
packetLength	CHOICE		EnumRange (AUTO NORMAL SHORT)
programNo	INTEGER		IntegerRange (-1 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
actualProgram	TEXT	R/O	StringRange R/O
video.colorBars	CHOICE		EnumRange (OFF ON)
video.barsType	CHOICE		EnumRange (BARS BLACK BLUE RED MAGENTA GREEN CYAN YELLOW WHITE)
video.genlockMode	CHOICE		EnumRange (OFF SLAVE REMOTE)
video.genlockHorzAdj	INTEGER		IntegerRange (0 .. 0)
video.genlockVerzAdj	INTEGER		IntegerRange (0 .. 0)
video.genlockPixAdj	INTEGER		IntegerRange (0 .. 64)
video.genlockCvbs	CHOICE		EnumRange (VIDEO SYNC)
video.genlockReset	TEXT		StringRange
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.embeddedAudio1	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio2	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio3	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio4	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio5	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio6	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)

name	*type*	*flags*	*range*
video.embeddedAudio7	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
video.embeddedAudio8	CHOICE		EnumRange (AUTO AUDIO1 AUDIO2 AUDIO3 AUDIO4 AUDIO5 AUDIO6 AUDIO7 AUDIO8 NONE)
audio.1.program	CHOICE		EnumRange (0 - OFF)
audio.1.delay	INTEGER		IntegerRange (-50 .. 800)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.program	CHOICE		EnumRange (0 - OFF)
audio.2.delay	INTEGER		IntegerRange (-50 .. 800)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.program	CHOICE		EnumRange (0 - OFF)
audio.3.delay	INTEGER		IntegerRange (-50 .. 800)
audio.3.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.program	CHOICE		EnumRange (0 - OFF)
audio.4.delay	INTEGER		IntegerRange (-50 .. 800)
audio.4.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.program	CHOICE		EnumRange (0 - OFF)
audio.5.delay	INTEGER		IntegerRange (-50 .. 800)
audio.5.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.program	CHOICE		EnumRange (0 - OFF)
audio.6.delay	INTEGER		IntegerRange (-50 .. 800)

name	*type*	*flags*	*range*
audio.6.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.program	CHOICE		EnumRange (0 - OFF)
audio.7.delay	INTEGER		IntegerRange (-50 .. 800)
audio.7.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.program	CHOICE		EnumRange (0 - OFF)
audio.8.delay	INTEGER		IntegerRange (-50 .. 800)
audio.8.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ebno	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.margin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E-UID1 BISS-E-UID2)
ca.bisskey	TEXT	NOCOMPARE	StringRange
ca.injectedid	TEXT	NOCOMPARE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.28 Advantech-ADAM5K

This page describes the device driver and the device window for the*Advantech ADAM5000/TCP* data acquisition unit. The device driver monitors and controls the digital inputs/outputs of various modules hosted by the ADAM5000/TCP unit. Analog or temperature sensor input functions are not covered by this driver, neither are counter/timer timer function of the some modules.

The *Advantech-ADAM5K* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Advantech-ADAM5K* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the*Advantech-ADAM5K* device driver as a low level driver to access the physical I/O ports.

Input ports

The device driver provides eight variables I00001, I00017, I00033, I00049, I00065, I00081, I00097 and I00113 which report the actual input state of digital input modules located in the slots 1..8 of the unit.

I00001 refers to the input ports of the module in the first slot (next to the ADAM5000/TCP communication module). The least significant bit of the variable denotes the state of DI0.

The variables report the physical state if the inputs. A bit is set to '1' if the input is open or externally driven to high level. The bit reads '0' if the input is connected to GND or externally driven to low level.

The device driver always reads the input state for all eight modules, regardles if there is an input module installed in a particular slot or not. This is because reading the information in this way gives the best performance. As a side effect, the driver displays some information for slots which contain other modules than digital inputs. This information should be ignored.

Output ports

The device driver provides eight variables Q00001, Q00017, Q00033, Q00049, Q00065, Q00081, Q00097 and Q00113 which report the output state of digital/realay output modules located in the slots 1..8 of the unit.

Q00001 refers to the output ports of the module in the first slot (next to the ADAM5000/TCP communication module). The least significant bit of the variable denotes the state of DO0.

The variables report the actual state of the outputs. Writing a bit to '1' closes the relay contact / opto-coupler. Reading a '1' reports a closed contact.

The device driver always reads the output state for all eight modules, regardless if there is an output module installed in a particular slot or not. This is because reading the information in this way gives the best performance. As a side effect, the driver displays some information for slots which contain other modules than digital outputs. This information should be ignored.

Output ports are read back by the driver after the sat-nms software has changed an output bit or every 10 seconds.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Advantech ADAM module acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the ADAM5K device in the device setup window , there must not be defined any other devices beside this module for this interface.
- The Advantech ADAM5K unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
I00001	HEX	R/O	HexRange R/O (0 .. 0)
I00017	HEX	R/O	HexRange R/O (0 .. 0)
I00033	HEX	R/O	HexRange R/O (0 .. 0)
I00049	HEX	R/O	HexRange R/O (0 .. 0)
I00065	HEX	R/O	HexRange R/O (0 .. 0)
I00081	HEX	R/O	HexRange R/O (0 .. 0)
I00097	HEX	R/O	HexRange R/O (0 .. 0)
I00113	HEX	R/O	HexRange R/O (0 .. 0)
Q00001	HEX		HexRange (0 .. 0)
Q00017	HEX		HexRange (0 .. 0)
Q00033	HEX		HexRange (0 .. 0)
Q00049	HEX		HexRange (0 .. 0)
Q00065	HEX		HexRange (0 .. 0)
Q00081	HEX		HexRange (0 .. 0)
Q00097	HEX		HexRange (0 .. 0)
Q00113	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
i.1	HEX	R/O	HexRange R/O (0 .. 0)
i.2	HEX	R/O	HexRange R/O (0 .. 0)
i.3	HEX	R/O	HexRange R/O (0 .. 0)
i.4	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
i.5	HEX	R/O	HexRange R/O (0 .. 0)
i.6	HEX	R/O	HexRange R/O (0 .. 0)
i.7	HEX	R/O	HexRange R/O (0 .. 0)
i.8	HEX	R/O	HexRange R/O (0 .. 0)
i.q	HEX	R/O	HexRange R/O (0 .. 0)

9.29 Advantech-ADAM6K

This page describes the device driver and the device window for the*Advantech ADAM6xxx* data aquisition module family. The device driver monitors and controls the digital inputs/outputs of various ADAM6xxx modules. Analog or temperature sensor input functions are not covered by this driver, neither are counter/timer timer function of the some modules. The driver can be used together with the following modules:

- ADAM-6018 (8 open collector outputs)
- ADAM-6050 (12 inputs, 6 outputs)
- ADAM-6051 (12 inputs)
- ADAM-6052 (8 inputs, 8 outputs)
- ADAM-6060 (6 inputs, 6 relay outputs)
- ADAM-6066 (6 inputs, 6 relay outputs)

The *Advantech-ADAM6K* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*AdvantechADAM6K* device driver directly (except to access the temperature readings). There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *SatService-IO-FEP* device driver as a low level driver to access the physical I/O ports.

Input ports

The variable I00001 refers to the input ports provided by the device. The least significant bit denotes the state of DI0.

The variable reports the physical state if the input. A bit is set to '1' if the input is open or externally driven to high level. The bit reads '0' if the input is connected to GND or externally driven to low level.

Output ports

The variable 'Q00001' reports the actual state of the digital outputs and also sets the output states. The least significant bit in the variable corresponds to output DO0.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Advantech ADAM module acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the device in the device setup window , there must not be defined any other devices beside this module for this interface.
- The Advantech ADAM modules are controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
I00001	HEX	R/O	HexRange R/O (0 .. 0)
Q00001	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

9.30 Advantech-Genesis

This page describes the device driver and the device window for the Advantech Wireless Genesis Series Amplifier. If more than one BUC is connected to the same Control panel the respective unit can be chosen from the device setup config.unit.

Note: Please be aware that in redundant setup all driver instances should communicate to the same common POC IP address. The HPA IPs are not reachable depending on switch position. The driver is designed to work for standalone units as well as in 1:1 redundancy. Other redundancy modes are not implemented and tested

It was tested with 1:1 redundant units type GenesisB1-250KuX

- SW version 1.2.02 and 1.2.04

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**

parameter	description
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
reference	Sets the reference sources of the BUC
maxAttn	Defines units max attenuation. Default setting is 30dB
lockLevel	Sets units lock level
unit	Defines the unit number (1..6) if more than one unit is controlled

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.unitType	TEXT	R/O	StringRange R/O
info.unitModel	TEXT	R/O	StringRange R/O
info.unitDescription	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.redUnit	TEXT	R/O	StringRange R/O
info.redMode	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.reference	CHOICE	SETUP	EnumRange (AUTO INTERNAL EXTERNAL)
config.maxAttn	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lockLevel	CHOICE	SETUP	EnumRange (OFF LEVEL-1 LEVEL-2 LEVEL-3)
config.unit	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
sys.tx.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
sys.tx.attn	FLOAT	R/O	DoubleRange R/O (0.0 .. 30.0)
sys.tx.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 30.0)
tx.alc	CHOICE		EnumRange (OFF ON SYS.INT-ON SYS-EXT-ON)
tx.agc	CHOICE		EnumRange (OFF ON)
tx.requestedLevel	FLOAT		DoubleRange (0.0 .. 0.0)
state	TEXT	R/O	StringRange R/O
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.humidity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.psuAuxVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
reset	TEXT	NOPRESET	StringRange
clearLatched	TEXT	NOPRESET	StringRange
clearFaultsTable	TEXT	NOPRESET	StringRange
protection	CHOICE		EnumRange (AUTO MANUAL LOCKED)

name	*type*	*flags*	*range*
standbyMode	CHOICE		EnumRange (HOT COLD WARM)
location	CHOICE	R/O	EnumRange R/O (STANDBY ONLINE)
position	CHOICE		EnumRange (A B)
online	TEXT	NOPRESET	StringRange
internal.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.faults	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.pref	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Forward RF/ALC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Input RF)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Cooling)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Switch)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Pre-amp/Input Driver)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Post amp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Auxiliary / IO Board)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature/Humidity)
faults.13	ALARM	R/O	AlarmFlagRange R/O (External Mute)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Fault Mute)
faults.15	ALARM	R/O	AlarmFlagRange R/O (TX Enable)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Int Bus los)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.31 Advantech-SSPB-GCP

This page describes the device driver and the device window for the Advantech Wireless BUC via GCP (General Control panel). The SSPB itself does not provide a Ethernet interface. Therefore the driver talks with the GCP via SNMP. If more than one BUC is connected to the same Control panel the respective unit can be chosen from the device setup config.unit.

It was tested with non redundant ARMA type 160W HPAs

- SW version 57

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The meter readings.
- --- The redundancy switching parameters.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- unit --- Defines the unit if more than one HPA is connected towards the GCP
- redundancy --- Defines if the unit is part of a 1:1 or 2:1 redundancy managed by the GCP or used as Standalone
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- accessMode --- Switch between local and remote access
- accessLock --- Lock unit
- extRef --- Switch between internal and external reference if supported
- reference --- Switch between auto and manual reference source selection
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.powerClass	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.serialCtrl	TEXT	R/O	StringRange R/O
info.shroudTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltagePsu1Ctrl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltagePsu2Ctrl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.elapsedTime	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.unit	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1 2:1)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.accessMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.accessLock	CHOICE	SETUP	EnumRange (UNLOCKED LOCKED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.reference	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 16.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 16.0)
tx.alc	CHOICE		EnumRange (ON OFF)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.hotspotTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange (MANUAL AUTO)

name	*type*	*flags*	*range*
location	CHOICE	R/O	EnumRange R/O (STANDBY ONLINE)
position	CHOICE		EnumRange DIS (A B)
online	TEXT	NOPRESET	StringRange DIS
onlineA	TEXT	NOPRESET	StringRange DIS
onlineB	TEXT	NOPRESET	StringRange DIS
reset	TEXT	NOPRESET	StringRange
internal.min	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.max	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.on	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.pref	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Units Positions)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transmit shutdown)
faults.03	ALARM	R/O	AlarmFlagRange R/O (No Communication)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TX Fault)
faults.05	ALARM	R/O	AlarmFlagRange R/O (TX Alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reflected Power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (ALC High)
faults.08	ALARM	R/O	AlarmFlagRange R/O (ALC Low)
faults.09	ALARM	R/O	AlarmFlagRange R/O (TX Switch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (TX SW Connected)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.32 Advantech-SSPB-MCP

This page describes the device driver and the device window for the Advantech Wireless BUC via a Monitor and control panel M&C 100 over serial interface. It uses the Advantech MCP protocol.

- Default TCP communication port is 10001

Default serial addresses are as following for 2:1 redundancy

- SSPA-A = 18
- SSPA-B = 19
- SSPA-S = 20

It was tested with 2:1 redundant SSPAs type: SSPA-ARMA-Cx400

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The meter readings.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	Device address for serial communication
Serial	Remember to set the line parameters for the serial interface
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.powerClass	INTEGER		IntegerRange (0 .. 0)
info.band	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
info.unit	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
tx.on	CHOICE		EnumRange (OFF ON)
tx.attn	FLOAT		DoubleRange (0.0 .. 16.0)
tx.attn.1	FLOAT		DoubleRange DIS (0.0 .. 16.0)
tx.attn.2	FLOAT		DoubleRange DIS (0.0 .. 16.0)
tx.alc	CHOICE		EnumRange (OFF ON)
tx.requestedLevel	FLOAT		DoubleRange (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.hotspotTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
protection	CHOICE		EnumRange (MANUAL AUTO)
location	CHOICE		EnumRange (STANDBY ONLINE)
position.1	TEXT	NOPRESET	StringRange DIS
position.2	TEXT	NOPRESET	StringRange DIS
reset	TEXT	NOPRESET	StringRange
internal.min	FLOAT		DoubleRange (0.0 .. 0.0)
internal.max	FLOAT		DoubleRange (0.0 .. 0.0)
internal.forwardPower	TEXT	R/O	StringRange R/O
internal.on	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.pref	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (PA Device)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fault shutdown)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reflected Pwr)
faults.05	ALARM	R/O	AlarmFlagRange R/O (ALC Suspended)
faults.06	ALARM	R/O	AlarmFlagRange R/O (ALC High)
faults.07	ALARM	R/O	AlarmFlagRange R/O (ALC Low)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Switch 2)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.33 Advantech-SSPB-SNMP

This page describes the device driver and the device window for the Advantech Wireless BUC over Ethernet interface (SNMP).

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.powerClass	INTEGER		IntegerRange (0 .. 0)
info.redundancy	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 16.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 16.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange DIS (STANDBY ONLINE)

name	*type*	*flags*	*range*
internal.min	FLOAT		DoubleRange (0.0 .. 0.0)
internal.max	FLOAT		DoubleRange (0.0 .. 0.0)
internal.forwardPower	TEXT	R/O	StringRange R/O
internal.on	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PA Device)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature Shut Down)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.34 Advantech-SSPB

This page describes the device driver and the device window for the Advantech Wireless BUC over serial interface.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	Device address for serial communication
Serial	Remember to set the line parameters for the serial interface (only if connected to serial port of beacon receiver and not to Ethernet)
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.extRef	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.powerClass	INTEGER		IntegerRange (0 .. 0)
info.redundancy	TEXT	R/O	StringRange R/O
info.band	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 16.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange DIS (STANDBY ONLINE)
position	CHOICE		EnumRange DIS (A B)
onSwitch	CHOICE	R/O	EnumRange R/O DIS (attached as A attached as B)
reset	TEXT	NOPRESET	StringRange
internal.min	FLOAT		DoubleRange (0.0 .. 0.0)
internal.max	FLOAT		DoubleRange (0.0 .. 0.0)
internal.forwardPower	TEXT	R/O	StringRange R/O
internal.on	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.pref	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PA Device)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Temperature Alarm)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature Pre Alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.35 Agilent-Spectrum-Analyzer

This page describes the device driver and the device window for HP / Agilent ESA spectrum analyzers. Actually the driver supports Agilent ESA E44xx analyzers equipped with a serial (RS232) interface. It provides a remote control of the basic parameters of the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- This page gives you access to the first 16 instrument memories of the spectrum analyzer. Clicking to 'STO x' stores the actual instrument settings to this memory place in the device. Clicking to 'RCL x' recalls the selected memory.You may use the text field between the buttons to note a comment for this particular memory place. Unless the instrument settings which are stored in the device itself, the text is stored at the VLC/M&C system. Hence, if you replace the spectrum analyzer, the stored instrument settings are gone with the replaced device, the comments however are still present.Also this page is available as a separate window. click to 'PRE' at the upper left corner of the spectrum view to launch a 'preset' window without leaving the spectrum view.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

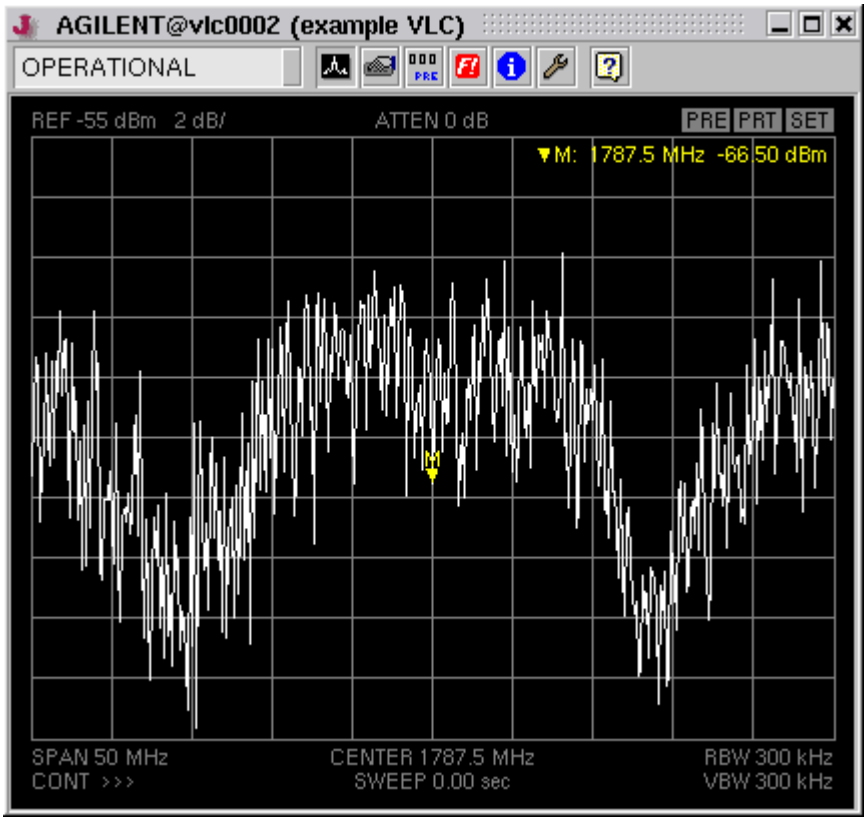
At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

- *parameter --- description*
- address --- The spectrum analyzer remote protocol does not use a device address. Leave this field empty.
- display --- This configuration parameter switches the local display of the spectrum analyzer on or off. Switching off the display improves the measurement speed of the analyzer in some situations.
- Serial --- The serial line parameters of interface used to drive the spectrum analyzer must be set to the same values set at the device itself (Accessible at the front panel using the 'System' menu). Regardless of the speed, RTS/CTS handshaking must be enabled and a full-wired RS232 cable must be used to the connect the spectrum analyzer to the M&C/VLC.The recommended line speed for the interface depends on the situation where the spectrum analyzer is used. Depending on the speed of the M&C/VLC system and the line speed used to drive the spectrum analyzer there results a maximum sweep rate the system can handle. This sweep rate must not exceed the capacity of the communication line to the NMS or the user interface. The recommended line speeds listed below restrict the sweep rates to values which avoid communication buffer overflows. | *environment* | *line speed* | *max. sweep rate* | | ---- | ---- | ---- | | VLC, connected to the NMS via modem dial up | *9600,E,8,1* | 0.3 / sec | | VLC, connected to the NMS via ISDN dial up or LAN | *19200,E,8,1* | 1 / sec | | M&C system, clients are connected via LAN or ISDN |*38400,E,8,1* | 2 / sec |

See also the note about sweep handling.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page.
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first

place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed an the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.manufacturer	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
invalidate.mkr	TEXT	R/O	StringRange R/O
invalidate.lev	TEXT	R/O	StringRange R/O
invalidate.frq	TEXT	R/O	StringRange R/O
invalidate.swp	TEXT	R/O	StringRange R/O
invalidate.oth	TEXT	R/O	StringRange R/O
invalidate.alw	TEXT	R/O	StringRange R/O
internal.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.lofr	FLOAT	R/O	DoubleRange R/O (0 .. 0)
config.display	CHOICE	SETUP	EnumRange (OFF ON)
config.bandEdge	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.loFreqLow	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.loFreqHigh	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT		StringRange
span.dn	TEXT		StringRange
span.up	TEXT		StringRange
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT		StringRange
reflev.dn	TEXT		StringRange

name	*type*	*flags*	*range*
scale	FLOAT		DoubleRange (0 .. 0)
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT		StringRange
rbw.dn	TEXT		StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT		StringRange
vbw.dn	TEXT		StringRange
sweep	FLOAT		DoubleRange (0.004 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT		StringRange
marker.nextPeak	TEXT		StringRange
marker.toCenter	TEXT		StringRange
marker.toReflev	TEXT		StringRange
reset	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
store	TEXT		StringRange
recall	TEXT		StringRange
memname.0	TEXT	SAVE	StringRange
memname.1	TEXT	SAVE	StringRange
memname.2	TEXT	SAVE	StringRange
memname.3	TEXT	SAVE	StringRange
memname.4	TEXT	SAVE	StringRange
memname.5	TEXT	SAVE	StringRange
memname.6	TEXT	SAVE	StringRange
memname.7	TEXT	SAVE	StringRange
memname.8	TEXT	SAVE	StringRange
memname.9	TEXT	SAVE	StringRange
memname.10	TEXT	SAVE	StringRange
memname.11	TEXT	SAVE	StringRange
memname.12	TEXT	SAVE	StringRange
memname.13	TEXT	SAVE	StringRange
memname.14	TEXT	SAVE	StringRange
memname.15	TEXT	SAVE	StringRange
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

9.36 Agilis-BUC

This page describes the device driver and the device window for the Agilis integrated BUC / SSPA over SNMP

The driver was tested with

- ALB 129 series compact 80/100W Ku Band BUC in 1:1 redundancy and as standalone unit

The driver was tested with the following SW Releases:

- BUC: 102,109,116
- SSPA: 113,114,116

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- startup --- Defines the default startup condition for the BUCs transmit state.
- frequency --- Defines the used lo frequency if the unit is equipped with more than one upconverter frequency.

- bucNumber --- Defines the identifier of the BUC in a 1:1 redundancy 1 or 2

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.startup	CHOICE	SETUP	EnumRange (ON OFF)
config.frequency	CHOICE	SETUP	EnumRange (FIXED LO LO-1 LO-2)
config.bucNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.versionBuc	TEXT	R/O	StringRange R/O
info.versionSspa	TEXT	R/O	StringRange R/O
info.redundancy	CHOICE	R/O	EnumRange R/O (ON OFF)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
position	CHOICE	R/O	EnumRange R/O DIS (A B)
location	CHOICE		EnumRange DIS (ONLINE STANDBY)
internal.on	TEXT	R/O	StringRange R/O
internal.pout	TEXT	R/O	StringRange R/O
internal.pin	TEXT	R/O	StringRange R/O
internal.pref	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (LO Lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FET Bias)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reflected Power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Module)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.37 Altec-Puregas-P210

This page describes the device driver and the device window for*Altec-Puregas-P210* waveguide pressurizer and dehydrator via SNMP.

It was tested with P210WLP running on SW: 3.11

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lowPressureThd	Defines the low pressure threshold in psi.
highPressureThd	Defines the high pressure threshold in psi.
highHumidityThd	Defines the high humidity threshold in %.
highRunTimeThd	Defines the high runtime threshold in seconds.
alarmDelay	Defines the alarm delay to be activated or not.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.

parameter	description
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `123456` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
internal.unitTemp	TEXT	R/O	StringRange R/O
internal.unitPres	TEXT	R/O	StringRange R/O
internal.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.lowPressureThd	FLOAT	SETUP	DoubleRange (0.3 .. 7.5)
config.highPressureThd	FLOAT	SETUP	DoubleRange (0.3 .. 7.5)
config.highHumidityThd	FLOAT	SETUP	DoubleRange (3.0 .. 15.0)
config.highRunTimeThd	INTEGER	SETUP	IntegerRange (120 .. 600)
config.hiDutyCycle	INTEGER	SETUP	IntegerRange (0 .. 99)
config.alarmDelay	CHOICE	SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tank.pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tank.pressure2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
lastRunTime	TEXT	R/O	StringRange R/O
totalRunTime	TEXT	R/O	StringRange R/O
dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.system	TEXT	R/O	StringRange R/O
state.compressor	TEXT	R/O	StringRange R/O
state.fan	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
runtime.reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (High outlet pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low outlet pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High humidity)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (High cabinet temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High duty cycle)
faults.06	ALARM	R/O	AlarmFlagRange R/O (High comp last run time)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Maintenace req)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Total alarm)

9.38 Altec-Puregas-P550

This page describes the device driver and the device window for*Altec-Puregas-P550* waveguide pressurizer and dehydrator via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
highFlowThd	Defines the high flow threshold value in ft2/h.
lowPressureThd	Defines the low pressure threshold in psi.

parameter	description
highPressureThd	Defines the high pressure threshold in psi.
highHumidityThd	Defines the high humidity threshold in %.
highRunTimeThd	Defines the high runtime threshold in seconds.
startupDelay	Defines the startup delay in seconds.
alarmDelay	Defines the alarm delay to be activated or not.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `123456` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.runTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.lastRunTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.highFlowThd	INTEGER	SETUP	IntegerRange (0 .. 900)
config.highPressureThd	FLOAT	SETUP	DoubleRange (0.3 .. 20.0)
config.lowPressureThd	FLOAT	SETUP	DoubleRange (0.3 .. 19.9)
config.highHumidityThd	INTEGER	SETUP	IntegerRange (3 .. 15)
config.highRunTimeThd	INTEGER	SETUP	IntegerRange (120 .. 300)
config.startupDelay	INTEGER	SETUP	IntegerRange (0 .. 10)
config.alarmDelay	CHOICE	SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pressure	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tank.pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
tank.pressure2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
humidity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flowRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
compressor	BOOLEAN	R/O	BooleanRange R/O (true false)
fan	BOOLEAN	R/O	BooleanRange R/O (true false)
heater	BOOLEAN	R/O	BooleanRange R/O (true false)
system	CHOICE	R/O	EnumRange R/O (ON SHUTDOWN)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (High Flow)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low pressure)
faults.04	ALARM	R/O	AlarmFlagRange R/O (High Humidity)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High Cab temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (High Compr. run time)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Maintenance required)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Total)

9.39 Anacom-KuBand-Transceiver

This page describes the device driver and the device window for the Anacom Ku-band transceiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- TX --- This page shows transmit part parameters of the device.
- RX --- This page shows receive part parameters of the device. If the device is configured to "TX-ONLY" mode, all parameters of this page remain gray.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Operational parameters

The transceiver's operational parameters contained on the TX/RX device window pages are mostly self explaining.

If you are using a unit with an external SSPA you should consider that the value shown as "measured output power" is a raw, uncalibrated reading which not necessarily has a linear relationship to the true output power of the device. This is is not a limitation of the sat-nms software but an issue with the transceiver itself.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Anacom transceiver.

parameter	***description***
address	You must set this parameter to the class/id (device address) set at the transceiver. The factory default for the device address id "FFFF", but you are encouraged to customize this value. See below for a description how to change the device address of a transceiver.
txOnly	The device driver has no possibility to detect if the transceiver is a transmit only type or a fully equipped unit. You must set this parameter to match the hardware you are using.
sspa	INTERN tells the device driver that the transceiver to control uses a built in amplifier. EXTERN modifies the behavior for transceiver models using an external SSPA unit.
powerOnDelay	You can force the transceiver to inhibit the transmitted signal for a certain time after power on. The time defined here only becomes effective if you set the next parameter to "ON".
warmUp	This parameter defined if you want to have the carrier forced off after power on for the time defined above. The special value "CANCEL" aborts an actually running warm up period.

parameter	***description***
faultMode	Defines the meaning of the fault relays of the transceiver. If you are using the transceiver together with an Anacom protection switch, set this parameter to "PROTECTION".
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. While the Anacom transceiver's factory preset setting is *1200,N,8,1* , the recommended line setting for the use with the sat-nms software is *9600,N,8,1* . See below how to change this setting at the transceiver.

Communication protocol / Interface type

This device driver requires the Anacom-Packet protocol. It is recommended to use an RS485 (4-wire RS422 mode) interface to control the transceiver. A single transceiver also may be controlled via RS232. If you are using an Anacom protection switch, however, the RS485 interface is mandatory.

Installation

Before you can operate an Anacom transceiver with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. For this, connect an ASCII terminal (or a PC running a terminal program) to the RS232 port of the transceiver. There comes a special cable with the transceiver for this purpose. Now perform the following steps:

1. Set the terminal line settings to 1200,N,8,1. Hit the "Return" key several times. The transceiver should report it's settings now.
2. Enter " *UTIMER OFF <RETURN>* " to switch off the automatic screen updates.
3. Enter " *R <RETURN>* ". The transceiver will answer something like "*R 123456 FFFF* ". The number 123456 is the unit's serial number, "FFFF" is the address it has actually assigned.
4. Set the new packet address. Enter "*R 123456 0101<RETURN>* ". You should replace 123456 by the serial number the transceiver actually reported. "0101" is the recommended device address for the first unit connected to one single interface. Subsequent units are addressed as "0102", "0103" and so on.
5. Now set the baud rate to 9600: "*BAUDRATE 9600 <RETURN>* ". The transceiver acknowledges this with some strange characters or not at all. This is because the transceivers baud rate no longer matches the rate test at the terminal.
6. Change the terminal's baud rate to 9600. Be sure not to press any key before the baud rate is 9600, the transceiver may fall back to 1200 baud if it does not recognize the keystrokes at 9600 baud.
7. Now finally enter "*SAVE <RETURN>* ". This permanently stores the settings you made. Without the "save" command the transceiver will forget all changes when it is powered off. The transceiver now is prepared for the usage with the satnms software. Remember/note the device address you have set at the device, it must be entered at the software at the maintenance page of the device window as described above.

Setting the EIRP alarm threshold

There is no parameter at the M&C user interface to set the "low EIRP" alarm threshold a the transceiver. The transceiver requires to be set in a special maintenance/programming mode to accept commands which change the alarm thresholds.

First of all, you need the *password* which enables the special programming mode at your transceiver. This password is unique for each transceiver, you have to ask Anacom for the password of your individual unit.

Then, the procedure to set the alarm threshold is slightly different for units with an integrated amplifier and for those with an external SSPA. In principle, there are three steps to:

1. Operate the transceiver at the threshold output power. Apply an input signal and adjust the transceiver gain and/or the drive level to achieve the output power which is just as low that an alarm shall be raised. You should measure the output power using a external power meter or in terms of EIRP by comparing the carrier with other signals you receive from the satellite.
2. Get the detector voltage, the transceiver measures for this output power. You will have the to set the threshold in raw ADC units, so the output power must be read in this way, too.
3. Program the new threshold into the transceiver.

To get the ADC reading, type the command ADC into the "low level command" field at maintenance page of the transceiver's device window. The transceiver should respond with something like **ADC FF 06 23 45 D4 00 00 00 6A .....** Each hexadecimal value in the list is the ADC reading for one measurement point the transceiver monitors. With a unit having an internal amplifier, the **second** value is the one of interest. For units with an external SSPA, note the **8th** value.

Now you can program this ADC value as the low output power threshold into the transceiver. For a transceiver with internal amplifier, type the following sequence of commands into the "low level command" field at maintenance page of the transceiver's device window (replace ?? by the threshold value to set). Wait a few seconds between entering the commands, the normal polling of the device state by the software goes on in background, so the commands are processed quite slow. .pre SPECIAL ART TXOUT ?? FF STORE TABLES EXIT .epre For a unit with external SSPA the command sequency is: .pre SPECIAL PA_LINES ON OFF OFF OFF OFF ON ART PA6 ?? FF STORE TABLES EXIT .epre Due to the fact, that a transceiver with an external SSPA uses the PA6 measurement point of read the detector voltage of the external SSPA, it will raise a "PA power supply" alarm instead of the "low TX power" alarm.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.txOnly	CHOICE	SAVE SETUP	EnumRange (TX-ONLY RX-TX)
config.sspa	CHOICE	SAVE SETUP	EnumRange (INTERN EXTERN)
config.powerOnDelay	INTEGER	SETUP	IntegerRange (0 .. 300)
config.warmUp	CHOICE	SETUP	EnumRange (OFF ON CANCEL)
config.faultMode	CHOICE	SETUP	EnumRange (NORMAL PROTECTION)
info.hw-revision	TEXT	R/O	StringRange R/O
info.sw-revision	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
rx.measuredLevel	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
rx.frequency	FLOAT		DoubleRange DIS (10950 .. 12750)
rx.gain	FLOAT		DoubleRange DIS (80.0 .. 105.0)
tx.measuredLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.frequency	FLOAT		DoubleRange (14000 .. 14500)
tx.gain	FLOAT		DoubleRange (38.0 .. 74.0)
tx.on	CHOICE		EnumRange (OFF ON)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
internal.flags.0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.flags.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.minGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.tx.maxGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Warmup delay)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.03	ALARM	R/O	AlarmFlagRange R/O (OSL lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TXL lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RXL lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TX output power)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TX input power)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Power supply 12V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Power supply 11V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply -5V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (OSL PLL voltage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TXL PLL voltage)
faults.14	ALARM	R/O	AlarmFlagRange R/O (RXL PLL voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply 5V)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply LNB)
faults.17	ALARM	R/O	AlarmFlagRange R/O (FEPROM)
faults.18	ALARM	R/O	AlarmFlagRange R/O (LNB)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Major alarm)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Minor alarm)

9.40 Anacom-Protection-Switch

This page describes the device driver and the device window for the Anacom 1:1 protection switch.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This is the main page to control the protection switch. The actual A/B position of the TX- and RX- switch is displayed separately, also the FIXED/AUTO state. Automatic protection is enabled if a switch shows "AUTO". Pressing the A/B buttons forces a switch to this position, reverting it to "FIXED" state. Press "AUTO" to re-enable automatic protection after you have selected the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Anacom protection switch.

parameter	*description*
address	You must set this parameter to the class/id (device address) set at the switch. The factory default for the device address id "FFFF", but as the switch shares it's communication interface with two transceivers it is necessary to customize this value. See below for a description how to setup the device addresses of a protection switch an the transceivers connected to it.
Serial	Remember to set the line parameters for the serial interface connected to the protection switch. While the Anacom protection switch' factory preset setting is *38400,N,8,1* , the recommended line setting for the use with the satnms software is *9600,N,8,1* . See below how to change this setting at the unit.

Communication protocol / Interface type

This device driver uses the Anacom-Packet protocol. It requires the switch to be controlled via

an RS485 (4-wire RS422 mode) interface. Operating the switch at the RS232 interface is principally possible, too, but then the transceivers controlled by the switch cannot be operated by the software. The "DIGIPEAT" mode of the switch is not supported.

Installation

Preparing a protection switch and the two transceivers for the usage with the sat-nms software is a little bit tricky, but with the description given below it should be no problem. To understand the following, consider that the protection switch and the two transceivers are controlled via a common RS485/RS422 bus. This requires the units to be set to the same baud rate, but to individual addresses. The description below uses the recommended addresses 0101 for transceiver A, 0102 for transceiver "B" and 0103 for the protection switch.

Preparing the transceivers is no problem, as they have a separate RS232 interface and they usually come with an RS232 cable you can use to program the transceivers before you connect them to the switch.

1. Program the transceiver "A" to the address "0101", as described on thehelp page for this device. Keep the transceiver disconnected from the protection switch for this procedure.
2. Repeat this for transceiver "B", however use the address "0102" fur this unit.
3. Connect the transceivers and the protection switch (i.e. your complete feed / outdoor equipment) and connect it to the M&C computer which already must be prepared to operate a redundant Anacom transceiver pair.
4. With everything connected and powered up, perform the following steps at the M&C user interface:
    - At the maintenance page of the protection switch' device window, set the device address "FFFF" and a line speed of 38400,N,8,1. After some seconds the protection switch should clear it's communication fault.
    - At the field "low level command" enter "*R* " to get the serial number of the switch. It will reply something like "R 123456 FFFF". Now enter " *R 123456 0103* " (you have to replace "123456" by the serial number actually reported by the device). The communication fault appears again.
    - Change the device address to "0103", this makes the communication fault disappear.
    - Enter at the "low level command" entry field "*BAUDRATE 9600* ".
    - Change the BAUDRATE at the M&C computer to 9600, too.
    - Enter the low level command " *DIGIPEAT OFF* ".
    - Finally enter the low level command "*SAVE* " to store the settings in the switch permanently.
5. Finish the setup by entering the device addresses of the transceivers at the maintenance page of their device windows.

The protection switch and the transceivers are now programmed to work with the sat-nms software.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.flags.0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.skip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.revision	TEXT	R/O	StringRange R/O
tx.pos	TEXT	R/O	StringRange R/O
tx.auto	TEXT	R/O	StringRange R/O
tx.setA	TEXT	NOPRESET	StringRange
tx.setB	TEXT	NOPRESET	StringRange
rx.pos	TEXT	R/O	StringRange R/O
rx.auto	TEXT	R/O	StringRange R/O
rx.setA	TEXT	NOPRESET	StringRange
rx.setB	TEXT	NOPRESET	StringRange
setAuto	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (WG switch)

9.41 Andrew-APC100

This page describes the device driver and the device window for the Andrew APC100 antenna controller.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.The set point parameters are shown with a resolution of 0.1 deg for azimuth/elevation and 1 deg for the polarization. This to make the spin buttons work in a convenient way. You still may enter azimuth/elevation values with a 0.01 deg resolution, the antenna controller accepts this.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The APC100 uses a one character device address in the range 'P' ... '_' (0x50 .. 0x5F). The factory default set by Andrew is 'P'.
polOffset	A non-zero value entered here is **subtracted** from each polarization reading before it is sent to the operator and **added** to each polarization setting before it finally is sent to the device. See the paragraph 'Polarization offset' below.
Serial	Remember to set the line parameters for the serial interface connected to the APC100. The APC100 expects *9600,E,8,1* .

Polarization offset

With the APC100, polarization angles always are positive. A typical range for the polarization setting is 0...180 deg. The sat-nms software however expresses the polarization setting as an offset to the ideal polarization setting for a satellite located at the same longitude as the antenna. These polarization offsets are in the range -90 deg .. +90 deg.

To adapt the APC100 polarization setting/reading to the sat-nms internal standard, the device driver introduces a 'polOffset' configuration parameter. Set this parameter to the polarization reading the APC100 shows while the antenna is positioned to azimuth=180 deg and horizontal polarization.

Software limit ranges

The APC100 provides min/max software limits for all three axes. These limits are set at the APC100 front panel using the "SETUP" menu. The device driver reads the limits from APC100 and forwards them to the sat-nms user interface. An operator will not be able to set antenna

pointing outside the limits programmed into the APC100.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.s1	HEX	R/O	HexRange R/O (0 .. 0)
internal.s2	HEX	R/O	HexRange R/O (0 .. 0)
internal.s3	HEX	R/O	HexRange R/O (0 .. 0)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.changed	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.polarization	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.polOffset	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0 .. 0)

name	*type*	*flags*	*range*
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Position data required)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Motor inhibit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Download limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Motor timeout)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Undefined position)

9.42 Antech-Deice

This page describes the device driver and the device window for the Antech De-icing controller.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
deice.mode	CHOICE		EnumRange (MANUAL AUTO)
deice.01	CHOICE		EnumRange (ON OFF)
deice.02	CHOICE		EnumRange (ON OFF)
deice.03	CHOICE		EnumRange (ON OFF)
deice.04	CHOICE		EnumRange (ON OFF)
state.ice	BOOLEAN	R/O	BooleanRange R/O (true false)
rainblower	CHOICE		EnumRange (ON OFF)
state.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Max. temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Blower)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Blower)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sector 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sector 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sector 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sector 4)

9.43 Antenna-BBU-Connector

The Antenna-BBU-Connector logical device controls three matrix switches to connect the TT&C signals of one antenna to a BBU device. These signals are two telemetry signals from the antenna to the receiver inpüuts and one telecommand signal from the BBU's modulator to the antenna.

The Antenna-BBU-Connector device is logically bound to one antenna. This may be either a movable antenna represented by an Antenna-Management device or a fixed antenna. In the latter case the name of the satellite the antenna is pointed to must be configured in the Antenna-BBU-Connector's setup.

The Antenna-BBU-Connector device permits to connect the antenna to a BBU from a given set of BBUs. The device offers only BBUs from this set, if they are capable to process the satellite this antenna is pointed to. This is either if the name of the satellite reported by the Antenna-Management device matches the satellite name configured for the particular BBU or if the BBU is marked to be a 'spare' one which may be used with any satellite.

Configuration

The configuration of the Antenna-BBU-Connector is defined in a configuration file. This file defines the list of available BBUs, the names of the matrix devices used for switching and the matrix ports the BBUs are connected to. The configuration file is mostly self explaining, here an example:

```
{
  "comment" : [ "Writing the comment this way",
              "this does not mess up a picky",
              "JSON parser" ],
  "antennas" : [
     {
        "device" : "AB-1",
        "satelliteId" : "AM-1.alias",
        "tcPort" : "01",
        "tm1Port" : "11",
        "tm2Port" : "12"
     },
     {
        "device" : "AB-2",
        "satellite" : "MY-SAT",
        "tcPort" : "02",
        "tm1Port" : "13",
        "tm2Port" : "14"
     }
  ],
  "bbus" : [
     {
        "name" : "BBU-1",
```

```
        "satellite" : "MY-SAT",
        "tcPort" : "11",
        "tm1Port" : "01",
        "tm2Port" : "01"
      },
      {
        "name" : "BBU-2",
        "satellite" : "MY-SAT",
        "tcPort" : "12",
        "tm1Port" : "02",
        "tm2Port" : "02"
      }
   ],
   "matrices" : [
      {
        "useFor" : "TC",
        "device" : "MTRX-TC"
      },
      {
        "useFor" : "TM1",
        "device" : "MTRX-TM1"
      },
      {
        "useFor" : "TM2",
        "device" : "MTRX-TM2"
      }
   ]
}
```

- *antennas:* Contains the list of antennas / Antenna-BBU-Connector devices this configuration file applies to. For each antenna the following parameters must be defined:
  - *device:* The name of the Antenna-BBU-Connector representing this antenna.
  - *satellite:* The name of the satellite the antenna points to. This parameter must be set for fixed antenna, for movable antennas it must be omitted.
  - *satelliteId:* The device name of the Antenna-Management device controlling this movable antenna. For fixed antennas this parameter has to be omitted.
  - *tcPort:* The matrix output delivering the telecommand signal to this antenna.
  - *tm1Port:* The matrix input this antenna feeds its telemetry 1 signal to.
  - *tm2Port:* The matrix input this antenna feeds its telemetry 2 signal to.
- *bbus:* Contains the list of BBUs which may be selected by the Antenna-BBU-Connector. For each BBU the following parameters must be defined:
  - *name:* The name of this BBU as shown in the selection drop down box. The name may be freely defined but should follow the restrictions applying to device names in the satnms software. It should consist of the characters A-Z, a-z, 0-9 and the '-' character only.
  - *satellite:* The name of the satellite this BBU is configured to. BBUs which may be

used with any satellite should have this set to "*".
    - *tcPort:* The matrix input receiving the telecommand signal from this BBU.
    - *tm1Port:* The matrix output delivering the telemetry 1 signal to this BBU.
    - *tm2Port:* The matrix output delivering the telemetry 2 signal to this BBU.
- *matrices:* Contains the list of matrices used to connected antennas and BBUs. This list must contain exactly three objects for the TC, TM1 and TM2 signals.
    - *useFor:* One of "TC", "TM1" or "TM2". Defines for which type of signal this matrix is used.
    - *device:* The device name of the matrix.

Remarks

- Satellite names must match exactly to make the Antenna-BBU-Connector work as expected. The satellite database, satellite names of fixed antennas and the satellite named defined this the BBUs must use exactly the same spelling.
- Matrix ports are to be defined in the way the matrix device driver expects them to identify the input / output parameter and setting. For most satnms matrix drivers this is a 2-digit decimal string with leading zeroes for the numbers 1 .. 9.
- The same matrix may be used to TM1 and TM2 signals. Nevertheless all three entries of the *matrices* list must be specified, defining the same device name for TM1, TM2.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the switch parameter to select a BBU for this antenna.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window if this device there is exactly one configuration parameter to configure the device:

parameter	***description***
confFile	The name of the configuration file defining antennas, BBUs and matrices

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.indication	TEXT	R/O	StringRange R/O
config.confFile	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (UNKNOWN)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch Combination)

9.44 Antenna-Dummy

This page describes the logical Antenna dummy device driver. It copies the azimuth, elevation and polarisation values to the actual state values.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The antenna pointing window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
azimuth	FLOAT	SAVE	DoubleRange (80.000 .. 280.000)
elevation	FLOAT	SAVE	DoubleRange (0.000 .. 90.000)
polarization	FLOAT	SAVE	DoubleRange (-110.000 .. 110.000)
trackMode	CHOICE	SAVE	EnumRange (OFF STEP ADAPTIVE)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.positioned	BOOLEAN	R/O	BooleanRange R/O (true false)

9.45 Antenna-Management

This page describes the device driver and the device window for the Antenna-Management logical device. This device lets you make a motorized antenna follow a satellite based on it's orbit position or ephemeris data. The Antenna-Management device is very similar to the Pointing-Robot device, extends this with a common database which stores the all satellite data like ephemeris data, stored center of box positions and administrative data of the satellite.

If used as a replacement for the legacy Antenna-Pointing device, the Antenna-Management device may be operated in "SIMPLE" mode, which reduces the number of parameter to focus the operation to a simple "mode the antenna to the stored position" for the selected satellite. See "Simple Mode" below for more information about this.

Using the device for recalling data from the database

To use the Antenna-Management device with the database, set the 'tracking data source' parameter to DATABASE. The workflow to track a satellite from the database is as follows:

1. Select the satellite from the 'satellite' list. You may enter partial satellite names to filter the list. There are several sort modes available for the satellite list, the list may be sorted by name, alias, orbit position and other satellite properties.

2. If the Antenna-Management device is setup to control a beacon receiver, select the satellite beacon to use.
3. If the Antenna-Management device is setup to control a TT&C equipment, select the telemetry and telecommand channels to use.
4. If you want to position or track following a tracking table, select the table file name set to use.
5. If you want to position or track following Intelsat I11 ephemerides, select the I11 data set to use.
6. Select the tracking mode to use (see below)
7. Click START to point the antenna to the satellite and to start tracking

The Antenna-Management device offers several tracking modes:

tracking mode	**description**
POINTING	The antenna moves to the pointing stored for the selected satellite in the database and stops.
STEPTRACK	The antenna moves to the pointing stored for the selected satellite in the database and then activates step track at the antenna controller.
MONOPULSE	The antenna moves to the pointing stored for the selected satellite in the database and then activates monopulse tracking at the antenna controller.
TLE	The antenna moves to the pointing calculated from the satellites TLE ephemerides and then tracks the satellite using this data.
I11	The antenna moves to the pointing calculated from the selected I11 ephemerides and then tracks the satellite using this data.
TABLETRACK	The antenna moves to the pointing calculated from the selected tracking table and then tracks the satellite using this data.
TLE-STEP	The antenna moves to the pointing calculated from the satellite's TLE ephemerides in the database and then activates step track at the antenna controller.
I11-STEP	The antenna moves to the pointing calculated from the selected I11 ephemerides in the database and then activates step track at the antenna controller.
TABLE-STEP	The antenna moves to the pointing calculated from the selected tracking table and then activates step track at the antenna controller.
TLE-MONOPULSE	The antenna moves to the pointing calculated from the satellite's TLE ephemerides in the database and then activates monopulse tracking at the antenna controller.

tracking mode	description
I11-MONOPULSE	The antenna moves to the pointing calculated from the selected I11 ephemerides in the database and then activates monopulse tracking at the antenna controller.
TABLE-MONOPULSE	The antenna moves to the pointing calculated from the selected tracking table and then activates monopulse at the antenna controller.

To stop tracking again click the STOP button.

Simple Mode

In SIMPLE mode only the tracking modes POINTING and STEPTRACK are available. The operation sequence differs from that used in FULL mode: First you select the desired tracking mode, then the satellite. The antenna moves to stored pointing for the satellite as soon as the satellite is selected from the list. No additional click to "start" is necessary.

"actual satellite" displays the satellite which recently has been selected for positioning and/or tracking. This persists over a restart of the sat-nms software.

Tracking file handling

The satellite database stores file names for table tracking. These names are assigned to a pair of satellite and antenna. During the satellite selection process the desired file name may be selected using the 'table tracking file name' parameter. The physical file however is **not** read from the database, it is read from the ~/robots directory ad the M&C computer. If the selected file is not found there, the Antenna-Management device complains with an error message.

Controlling the antenna without using the database

To use the TLE / I11 / table Track capabilities of the Antenna-Management device for a satellite which is not stored in the database or has no valid ephemeris data stored there, set the 'tracking data source' parameter to MANUAL and enter the tracking data at the 'Pointing Robot' page. The typical workflow there is to copy the ephemeris data from e.g. a webbrowser window and paste it into the 'ephemeris data' field at this page. The paragraphs below describe the various formats the Antenna-Management device can interpret.

Move the antenna along a satellite position calculated from ephemeris data

You can make the pointing robot follow a satellite position computed from usual ephemeris data sets. The robot recognizes the following data formats:

- **Keplerian elements** --- The robot recognizes and parses keplerian elements in the NASA 2-line format. The two lines of element definition may be prepended by an additional line with the satellite's name. A common source for Keplerian elements in this format is www.celestrak.com . The pointing robot uses the NORAD SDP4 model to predict a satellite's position from this type of data.
- **Intelsat elements** --- Intelsat publish ephemeris data for their communication satellites

at their web site. The data format and the mathematical model for this data is Intelsat proprietary. The pointing robot recognizes this data type as well. You should copy the complete text of the ephemeris data set into the entry field to make the software recognize the data correctly.

Pointing the antenna to the nominal position of a satellite

If a single floating point number is entered to the ephemeris data field, the pointing robot interprets this as a nominal orbit position of the satellite (°E).

Move the antenna along a path given by a table

The pointing robot is capable to move the antenna along a path defined in a text file table. You either may copy the file to the IDU computer by means of FTP and enter the name of file to the ephemeris field or you may enter or cot/paste the table directly to this field. If you do the latter, the size of the table is limited to 32000 characters.

The Pointing-Robot provides a file selection dropdown widget to select files located in the /home/satnms/robot directory on the M&C&VLC server. To use this feature, select the file from the dropdown list and then click "SELECT". This copies the file name into the ephemeris data field and thus selects the file.

Files in the /home/satnms/robot directory which are no longer used may be deleted by selecting the file do delete from the drop down list and clicking DELETE. A file actually in use cannot be deleted, the request is silently ignored.

Polarization calculation

When calculating the antenna pointing from a orbital model or when interpolating the pointing from the table data, the Pointing-Robot also calculates a polarization angle. As each satellite uses its individual polarization offset, the calculated angle is normally for information only. With table tracking, the pointing robot may also command the calculated polarization angle at the antenna.

Setting "apply pol" to CALCED enables this feature. Setting "apply pol" to SAT-DB makes the devices use the polarization angle stored for this satellite in the database.

Tracking table format

The format of the table must be as follows: It contains the data as a three or four column table. Empty lines are ignored, comments starting with a '#' as well. The numbers in the table are parsed as floating point numbers which only may consist of decimal digits, one decimal point and an optional leading '-' if negative. The columns must be separated by an arbitrary number of space or tabulator characters. They have the following meanings:

- 1 --- time stamp The time stamp must be a Julian date with the time of day coded as a fraction of a day. Example: The Julian date for the common base of most computer clocks (1970-01-0100:00:00) is 2440587.5
- 2 --- azimuth angle (degrees, 0.0 == north)
- 3 --- elevation angle (degrees)

Beside this format, the Antenna-Management device accepts MMS files with the following column definitions:

- 1 --- date in the format YY/MM/DD
- 2 --- time in the format HH:MM:SS.sss (UTC)
- 3 --- elevation angle (degrees)
- 4 --- azimuth angle (degrees, 0.0 == north)

File import

The Pointing Robot GUI offers an import function which permits to copy pointing tables from the client computer to the server and thereby convert the file format to that one internally used by the sat-nms software.

Parsing order

If the pointing robot gets entered a new ephemeris data set, it tries to interpret the data in the following way:

1. It checks if the first line of the data specifies an existing file (residing on the IDU computer, therefore the file path should use Unix-style ('/') directory separators if necessary. If there is the specified file, it gets read an interpreted in a way, its file extension suggests: '*.tle' or '*.TLE': NASA 2-line ephemeris data. '*.i11' or '*.I11': Intelsat ephemeris data. Files with other extensions are interpreted as table data as described above.
2. It checks if the data is a valid NASA 2-line data set. A satellite name before the 2-line data is accepted and parsed separately if present.
3. It checks if the data contains valid Intelsat ephemeris data.
4. It checks if the first number in the first line of the data is a valid Julian date in the range 2005 .. 2050. In this case the robot assumes, that the ephemeris data contain a pointing table.
5. Finally, if nothing of the above mentioned formats did match, the robot tries to interpret the first line of the data as the nominal orbit position of the satellite.

If all of the parsing attempts fail, the pointing robot reports the model type to be UNKNOWN. Mostly this happens due to artifacts of previously entered elements which had not been deleted before the new ephemeris data was entered / pasted.

Remote control

The Pointing-Robot may be updated easily with updated pointing tables or ephemeris data by copying such files to the /home/public directory of the M&C system using the SMB file sharing protocol and then commanding the name of this file to the 'elements.input' variable of the Pointing-Robot using one of the M&C system's remote control ports.

ACU fault monitoring

The device can monitor the ACU's summary fault and show it in the device screen. The parameter 'antenna fault detection' may be set to one of the following settings:

- **IGNORE-ACU-FAULT** : The pointing robot ignores ACU faults, does not show them.
- **SHOW-ACU-FAULT** : The pointing robot shows ACU faults, continues to command the antenna controller even if there is an ACU fault.
- **STOP-ON-ACU-FAULT** : The pointing robot stops when it detects a fault of the antenna controller. It must be restarted manually after the ACU fault has been cleared.

Fault flags

In table tracking mode the Pointing-Robot reports two fault conditions regarding the end of table data:

File expired is reported if the end of the table data is reached.

Some hours before (configurable in the device setup) the**File expires soon** fault is reported. The latter fault condition appears as a WARNING by default.

Both fault conditions are reported if the Pointing-Robot is in table tracking mode, even if the tracking is stopped. To clear faults concerning an outdated but unused table, simply clear the "ephemeris data" field.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page lets you select a satellite and make the antenna track along the data read from the database for the selected satellite.
- --- The "Pointing Robot" page lets you manually enter ephemeris data for the satellite to track and control the Pointing-Robot integrated into the Antenna-Management directly.
- --- The "Create Robot File" lets you create a table tracking file from ephemeris date read from the database for a given time span for offline use.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
azId	The parameter ID used to control the antenna's azimuth angle.
elId	The parameter ID used to control the antenna's elevation angle.

parameter	*description*
polId	The parameter ID used to control the antenna's polarization angle.

```
    // the connection may be closed meanwhile
```

| azStateId | The parameter ID reporting the antenna's actual azimuth angle. | | elStateId | The parameter ID reporting the antenna's actual elevation angle. | | polStateId | The parameter ID reporting the antenna's actual polarization angle. | | posReachedId | A parameter ID reporting the antenna's 'position reached' state as a boolean parameter. | | motorStopId | The parameter ID used to issue a "motor stop" at the antenna. Leave this empty if the antenna controller does not support this | | motorStopVal | The value to be sent to the parameter ID above in order to stop the motors | | memoryClearId | The parameter ID used to make the antenna clear its tracking memory. Leave this empty if the antenna controller does not support this | | memoryClearVal | The value to be sent to the parameter ID above in order to clear the tracking memory | | stepTrackId | The parameter ID used to switch the antenna controller's step track ON/OFF. Leave this empty if the antenna controller does not support step track | | stepTrackOn | The value to be sent to the parameter ID above in order to enable step track at the antenna controller | | stepTrackOff | The value to be sent to the parameter ID above in order to disable step track at the antenna controller | | targetId | This is a special configuration parameter for using the Antenna-Management device with a Vertex 7200 ACU. This ACU requires a target number to be set in order to run a step track. Set this to '.selectTarget' when controlling a Vertex-7200 ACU. | | monopulseTrackId | The parameter ID used to switch the antenna controller's monopulse tracking ON/OFF. Leave this empty if the antenna controller does not support monopulse tracking | | monopulseTrackOn | The value to be sent to the parameter ID above in order to enable monopulse tracking at the antenna controller | | monopulseTrackOff | The value to be sent to the parameter ID above in order to disable monopulse tracking at the antenna controller | | targetNameId | If set, the Antenna-Management device sends the name and orbit position of the satellite actually tracked to this parameter. This can be used, to display the actually tracked satellite in the device screen of an ACU supporting such a parameter | | beacon.freqId | The parameter ID used to control the beacon receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a beacon receiver | | beacon.attenId | The parameter ID used to control the beacon receiver's attenuation. The parameter value is sent in dB as a floating point value | | beacon.levelId | The parameter ID reporting the beacon level. The parameter addressed by this ID should report the level as a dB value (either absolute or relative) | | beacon.polId | The parameter ID used to control the beacon polarization pane. | | tc.freqId | The parameter ID used to control the telecommand transmit frequency. The value sent is the transmit frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a telecommand transmitter | | tc.attenId | The parameter ID used to control the telecommand transmitter attenuation. The parameter value is sent in dB as a floating point value | | tc.polId | The parameter ID used to control the telecommand signal polarization pane. | | tm1.freqId | The parameter ID used to control the primary telemetry receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the

Antenna-Management device shall not control a telemetry receiver | | tm1.attenId | The parameter ID used to control the primary telemetry receiver attenuation. The parameter value is sent in dB as a floating point value | | tm1.polId | The parameter ID used to control the primary telemetry receiver polarization pane. | | tm2.freqId | The parameter ID used to control the secondary telemetry receiver's frequency. The value sent is the receive frequency in MHz as the true RF frequency of the signal. Leave this empty if the Antenna-Management device shall not control a telemetry receiver | | tm2.attenId | The parameter ID used to control the secondary telemetry receiver attenuation. The parameter value is sent in dB as a floating point value | | tm2.polId | The parameter ID used to control the secondary telemetry receiver polarization pane. | | reserved | Used by a RX-Channel device to lock the antenna. | | reserved.lo.x | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.lo.y | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.hi.x | Used by a RX-Channel device to lock one RX band of the antenna. | | reserved.hi.y | Used by a RX-Channel device to lock one RX band of the antenna. | | beamwidth | The antenna's half 3dB beam width in °. | | diameter | The antenna diameter in m. | | latitude | The antenna's geodetic location (latitude, °N). | | longitude | The antenna's geodetic location (longitude, °E). | | altitude | The antenna's altitude over sea level in m. | | ephemeris.update | If this is set to MANUAL, TLE and I11 data sets are read once from the database when tracking is started. If set to AUTO, TLE and I11 parameter are updated automatically while tracking is active and the data gets updated in the database. |

Database configuration

To give the Antenna-Management device access to the satellite database, the database server IP address as well as user name and password must be defined in the vlc.properties file. The property keys used for this are:

```
db.satlist.host=<IP4-address>
db.satlist.user=<user>
db.satlist.password=<password>
db.satlist.usessl=false
```

If SSL encryption shall be used for the database communication, set the #usessl' key to 'true'. In this case the M&C application's home directory must contain a hidden '.postgresql' directory with the database server's certificate, key and root certificate files. These files can be obtained from the database administrator, they have to be named postgresql.crt, postgresql.key.der and root.crt.

Changes to the vlc.properties file become effective after the next M&C restart.

Remarks

The accuracy of the calculations made by Pointing-Robot depend on the following:

- The geodetic location of the antenna must be known very precisely.
- The antenna's azimuth axis alignment must be precisely perpendicular.
- The clock of the M&C computer must be set to UTC.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.azId	TEXT	SAVE SETUP	StringRange
config.elId	TEXT	SAVE SETUP	StringRange
config.polId	TEXT	SAVE SETUP	StringRange
config.azStateId	TEXT	SAVE SETUP	StringRange
config.elStateId	TEXT	SAVE SETUP	StringRange
config.polStateId	TEXT	SAVE SETUP	StringRange
config.posReachedId	TEXT	SAVE SETUP	StringRange
config.stepTrackId	TEXT	SAVE SETUP	StringRange
config.stepTrackOn	TEXT	SAVE SETUP	StringRange
config.stepTrackOff	TEXT	SAVE SETUP	StringRange
config.targetId	TEXT	SAVE SETUP	StringRange
config.targetNameId	TEXT	SAVE SETUP	StringRange
config.monopulseTrackId	TEXT	SAVE SETUP	StringRange
config.monopulseTrackOn	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.monopulseTrackOff	TEXT	SAVE SETUP	StringRange
config.motorStopId	TEXT	SAVE SETUP	StringRange
config.motorStopVal	TEXT	SAVE SETUP	StringRange
config.clearMemoryId	TEXT	SAVE SETUP	StringRange
config.clearMemoryVal	TEXT	SAVE SETUP	StringRange
config.beacon.freqId	TEXT	SAVE SETUP	StringRange
config.beacon.levelId	TEXT	SAVE SETUP	StringRange
config.beacon.attenId	TEXT	SAVE SETUP	StringRange
config.beacon.pollId	TEXT	SAVE SETUP	StringRange
config.tc.freqId	TEXT	SAVE SETUP	StringRange
config.tc.atten1Id	TEXT	SAVE SETUP	StringRange
config.tc.atten2Id	TEXT	SAVE SETUP	StringRange
config.tc.pollId	TEXT	SAVE SETUP	StringRange
config.tm1.freqId	TEXT	SAVE SETUP	StringRange
config.tm1.attenId	TEXT	SAVE SETUP	StringRange
config.tm1.pollId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.tm2.freqId	TEXT	SAVE SETUP	StringRange
config.tm2.attenId	TEXT	SAVE SETUP	StringRange
config.tm2.polId	TEXT	SAVE SETUP	StringRange
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.ephemeris.update	CHOICE	SAVE SETUP	EnumRange (MANUAL AUTO)
config.uiMode	CHOICE	SAVE SETUP	EnumRange (FULL SIMPLE)
config.fixedAntenna	CHOICE	SAVE SETUP	EnumRange (NO YES)
dataSource	CHOICE	SAVE	EnumRange (DATABASE MANUAL)

name	*type*	*flags*	*range*
satsort	CHOICE	SAVE	EnumRange (NAME ALIAS ORBIT NORAD FLIGHT)
reconnect	TEXT	NOPRESET	StringRange
poslist	CHOICE		EnumRange (?)
satlist	CHOICE		EnumRange (?)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
noradNumber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
intDesignator	TEXT	R/O	StringRange R/O
orbit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inclination	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
alias	TEXT	R/O	StringRange R/O
trackMode	CHOICE	SAVE	EnumRange (POINTING STEPTRACK MONOPULSE TLE I11 TABLETRACK TLE-STEP I11-STEP TABLE-STEP TLE-MONOPULSE I11-MONOPULSE TABLE-MONOPULSE)
beacon	CHOICE		EnumRange DIS (?)
tcchan	CHOICE		EnumRange DIS (?)
tmchan	CHOICE		EnumRange DIS (?)
i11	CHOICE		EnumRange (?)
table	CHOICE		EnumRange (?)
ephemeris	TEXT	R/O	StringRange R/O
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
reload	TEXT	NOPRESET	StringRange
gotoTLE	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
mayStart	BOOLEAN	R/O	BooleanRange R/O (true false)
mayStop	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
elevation	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
polarization	FLOAT	NOPRESET	DoubleRange (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.activeFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.latchedFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.usePol	TEXT	R/O	StringRange R/O
initial.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
initial.state	TEXT	R/O	StringRange R/O
satoperator.name	TEXT	R/O	StringRange R/O
satoperator.contact	TEXT	R/O	StringRange R/O
satoperator.exists	BOOLEAN	R/O	BooleanRange R/O (true false)
saved.satelliteid	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.i11id	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.tableid	INTEGER	SAVE	IntegerRange (0 .. 0)
saved.trackMode	INTEGER	SAVE	IntegerRange (0 .. 0)
tracking	CHOICE	SAVE	EnumRange (OFF ON)
interval	INTEGER	SAVE	IntegerRange (0 .. 3600)

name	*type*	*flags*	*range*
calculatePol	CHOICE	SAVE	EnumRange (OFF CALCED SAT-DB)
antFaultMode	CHOICE	SAVE	EnumRange (IGNORE-ACU-FAULT SHOW-ACU-FAULT STOP-ON-ACU-FAULT)
expireWarning	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
elements.input	TEXT		StringRange
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.orbit	TEXT	R/O	StringRange R/O
state.short	TEXT	R/O	StringRange R/O
state.long	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
state.satellite	TEXT	R/O	StringRange R/O
state.alias	TEXT	R/O	StringRange R/O
state.intDesignator	TEXT	R/O	StringRange R/O
state.noradNumber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackMode	TEXT	R/O	StringRange R/O
state.stepTrack	TEXT	R/O	StringRange R/O
offset.az	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.el	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.pl	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
hysteresis.az	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.el	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.pl	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
elements.type	TEXT	R/O	StringRange R/O
elements.age	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
elements.epoch	TEXT	R/O	StringRange R/O
elements.validFrom	TEXT	R/O	StringRange R/O
elements.validTo	TEXT	R/O	StringRange R/O
resetLatched	TEXT	NOPRESET	StringRange
files	CHOICE		EnumRange (NONE)
selectFile	TEXT	NOPRESET	StringRange
deleteFile	TEXT	NOPRESET	StringRange
forceUpdateFileList	TEXT	NOPRESET	StringRange
wf.startTime	TEXT		StringRange
wf.endTime	TEXT		StringRange
wf.step	INTEGER		IntegerRange (1 .. 7200)
wf.fileName	TEXT		StringRange
wf.write	TEXT	NOPRESET	StringRange
wf.mayWrite	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Database access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Antenna Controller)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Beacon Receiver)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input Device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (File expired)
faults.06	ALARM	R/O	AlarmFlagRange R/O (File expires soon)
faults.07	ALARM	R/O	AlarmFlagRange R/O (ACU Latched)

9.46 Antenna-Pointing-Fixed

The Antenna-Pointing-Fixed logical device is a variant of theAntenna-Pointing device for non-motorized antennas. Such (small) antennas may view multiple satellites at a time. The Antenna-Pointing-Fixed device permits to define these satellites in the satellites list of the device, thus making the antenna work together with RX-Channel or RX-LineSettings-CDB devices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing values, the selected satellite name and the orbit position of this satellite. You should use this page if you want to move the antenna to a certain orbit position.
- --- The list of satellite positions. You can select a satellite to move the antenna to with this page and you can maintain the satellite list.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Pointing device work properly.

parameter	***description***
config.reserved.lo.x	
config.reserved.lo.y	
config.reserved.hi.x	
config.reserved.hi.y	These settings give a hint to the RX-LineSettings-CDB devices, that a particular port of the antenna actually is reserved and not available for automatic antenna selection. These settings have no impact to the Antenna-Pointing device itself.
azId	The message ID controlling the antenna's azimuth parameter.
elId	The message ID controlling the antenna's elevation parameter.
polId	The message ID controlling the antenna's polarization parameter.
orbitId	The message ID controlling the antenna's orbit parameter.
polPlaneId	The message ID selecting the antenna's polarization plane. See section "Polarization plane handling" below.

parameter	*description*
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
beamwidth	The antenna's half 3dB beamwidth. A satellite is considered a pointed by the antenna, if the azimuth/elevation angles of the antenna match the stored pointing of a satellite within +/- beamwidth.

Operation modes

The Antenna-Pointing device knows three different modes of operation. This is to adapt the behaviour of the device to different types of antennas. By setting the azId, elId, polId and orbitId configuration parameter or leaving them empty you select the proper mode of operation for a particular antenna:

- **Normal 3-axis antenna** : For this antenna type, set the azId, elId and polId configuration parameters to the appropriate parameter identifiers. Leave the orbitId setting empty.In this operation mode, the device will control all three axes of the antenna and it will store complete (az/el/pol) pointing vectors in the satellite table. Only with this operation mode the device is capable to perform *orbit position* to *antenna pointing* calculations.
- **Polamount type antenna** : For this antenna type, only set the azId configuration setting. Leave the other message ID parameters empty.The device will use the azimuth angle to control the antenna's orbit arc setting. This value together with a orbit position angle gets stored in the satellite table.
- **Intelligent mobile SNG antenna** : The third mode of operation is used with car mobile SNG antenna controllers which accept a satellite orbit position angle to point the antenna to this satellite. To enable this mode, leave the azId, elId and polId configuration settings empty and set the orbitId parameter to the appropriate value.The device will act as a simple satellite name to orbit position translator in this mode. Only the orbit position gets stored in the satellite table, the orbit position will be commanded to the antenna controller when a satellite is selected.

Polarization plane handling

The Antenna-Pointing device accepts an operational parameter called 'rxpolplane' (H/V). This parameter may be used to select a certain receive polarization plane by turning the feed (This works with 3-axis antennas only.)

The device translates the commanded polarization angle when forwarding it to the antenna. If horizontal polarization is selected (which is the default if the parameter isn't used), the angle is forwarded as it is. With vertical polarization, the device either adds or subtracts 90 degrees before the value gets forwarded. The device decides if to add or to subtract based on the range of the polarization angle given by antenna controller driver.

If the configuration parameter polPlaneId is set, the device will forward the polarization plane setting to the given device / parameter rather than shifting the polarization angle. The destination may be a switch selecting the polarization plane or a comparable parameter of the antenna controller itself. This "pass-through" mode may be used with all types of antenna controllers, as long as there is a parameter to select the polarization plane.

Remarks

1. To make the antenna move to a satellite already stored in the list, double click to the satellite's entry in the - page.
2. Pressing the right mouse button on a satellite's entry in the list opens a context menu offering all functions to manage the satellite list.
3. When entering a new satellite to the list, first move the antenna to it's calculated position by entering the satellite's orbit position at the - page. Optimize the antenna pointing. Then switch to the - page, add a new entry to the list and update this entry with the current antenna pointing.

Satellite position file

The Antenna-Pointing device stores the satellite positions in a text file in the base directory of the M&C/VLC software. It is named "XXXX.txt" where XXXX is the name of the Antenna-Pointing logical device. The file contains the satellite list as a table of semicolon separated fields. It may be imported by common spread sheet programs this way. The table columns are defined as follows.

1. satellite name
2. orbit position (deg. east)
3. antenna azimuth
4. antenna elevation
5. antenna polarization offset

Be aware, that the numeric values are stored with a decimal dot (##.###), regardless of the default locale of your computer.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.azId	TEXT	SETUP	StringRange DIS
config.elId	TEXT	SETUP	StringRange DIS
config.polId	TEXT	SETUP	StringRange DIS
config.orbitId	TEXT	SETUP	StringRange DIS
config.polPlaneId	TEXT	SAVE SETUP	StringRange
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.satlist	GENERIC		ObjectRange generic object
config.longitude	FLOAT	SAVE SETUP	DoubleRange (-180.00 .. 180.00)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
update	TEXT	NOPRESET	StringRange
azimuth	FLOAT	NOPRESET	DoubleRange DIS (0.0 .. 0.0)
elevation	FLOAT	NOPRESET	DoubleRange DIS (0.0 .. 0.0)
polarization	FLOAT	NOPRESET	DoubleRange DIS (0 .. 0)
orbit	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
rxpolplane	CHOICE	SAVE NOPRESET	EnumRange (H V)
polamount	BOOLEAN	R/O	BooleanRange R/O (true false)
fixed	BOOLEAN	R/O	BooleanRange R/O (true false)
orbitPassThrough	BOOLEAN	R/O	BooleanRange R/O (true false)
locked	BOOLEAN	SAVE	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU Fault)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (ACU Warning)

9.47 Antenna-Pointing-Hiltron

The Antenna-Pointing-Hiltron logical device is a special version of theAntenna-Pointing device for use with the Hiltron-ACU2 device. Unlike the standard version of the device, this specialized version uses the target memories stored in the ACU as the satellite list rather than maintaining a separate list for this.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing values, the selected satellite name and the orbit position of this satellite. You should use this page if you want to move the antenna to a certain orbit position.
- --- The list of satellite positions. You can select a satellite to move the antenna to with this page and you can store or delete target definitions list.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Pointing device work properly.

parameter	*description*
config.reserved.lo.x	
config.reserved.lo.y	
config.reserved.hi.x	
config.reserved.hi.y	These settings give a hint to the RX-LineSettings-CDB devices, that a particular port of the antenna actually is reserved and not available for automatic antenna selection. These settings have no impact to the Antenna-Pointing device itself.
acuDevice	The device name of the ACU. This must be aHiltron-ACU2 device, the Antenna-Pointing-Hiltron device does not work with other antenna controllers.

parameter	*description*
beamwidth	The antenna's half 3dB beamwidth. A satellite is considered a pointed by the antenna, if the azimuth/elevation angles of the antenna match the stored pointing of a satellite within +/- beamwidth.
diameter	The antenna's dish diameter in meters. This is used by theRX-Channel device when sorting antennas by size in activation based mode.
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.

Polarization plane handling

The Antenna-Pointing device accepts an operational parameter called 'rxpolplane' (H/V). This parameter may be used to select a certain receive polarization plane by turning the feed .

The device translates the commanded polarization angle when forwarding it to the antenna. If horizontal polarization is selected (which is the default if the parameter isn't used), the angle is forwarded as it is. With vertical polarization, the device either adds or subtracts 90 degrees before the value gets forwarded. The device decides if to add or to subtract based on the range of the polarization angle given by antenna controller driver.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.acuDevice	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (-180.00 .. 180.00)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
azimuth	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
elevation	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
polarization	FLOAT	NOPRESET	DoubleRange (0 .. 0)
orbit	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
rxpolplane	CHOICE	SAVE NOPRESET	EnumRange (H V)
locked	BOOLEAN	SAVE	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU Fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (ACU Warning)

9.48 Antenna-Pointing-ODM

The Antenna-Pointing-ODM logical device is a special version of theAntenna-Pointing device for use with the SatService-ACU-ODM device. Unlike the standard version of the device, this specialized version uses the target memories stored in the ACU as the satellite list rather than maintaining a separate list for this.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing values, the selected satellite name and the orbit position of this satellite. You should use this page if you want to move the antenna to a certain orbit position.
- --- The list of satellite positions. You can select a satellite to move the antenna to with this page and you can store or delete target definitions list.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Pointing device work properly.

parameter	***description***
config.reserved.lo.x	
config.reserved.lo.y	
config.reserved.hi.x	
config.reserved.hi.y	These settings give a hint to the RX-LineSettings-CDB devices, that a particular port of the antenna actually is reserved and not available for automatic antenna selection. These settings have no impact to the Antenna-Pointing device itself.
odmDevice	The device name of the ACU/ODM. This must be aSatService-ACU-ODM device, the Antenna-Pointing-ODM device does not work with other antenna controllers.
beamwidth	The antenna's half 3dB beamwidth. A satellite is considered a pointed by the antenna, if the azimuth/elevation angles of the antenna match the stored pointing of a satellite within +/- beamwidth.
diameter	The antenna's dish diameter in meters. This is used by theRX-Channel device when sorting antennas by size in activation based mode.
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.

Polarization plane handling

The Antenna-Pointing device accepts an operational parameter called 'rxpolplane' (H/V). This parameter may be used to select a certain receive polarization plane by turning the feed .

The device translates the commanded polarization angle when forwarding it to the antenna. If horizontal polarization is selected (which is the default if the parameter isn't used), the angle is forwarded as it is. With vertical polarization, the device either adds or subtracts 90 degrees before the value gets forwarded. The device decides if to add or to subtract based on the range of the polarization angle given by antenna controller driver.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.odmDevice	TEXT	SAVE SETUP	StringRange
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (-180.00 .. 180.00)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
azimuth	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
elevation	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
polarization	FLOAT	NOPRESET	DoubleRange (0 .. 0)
orbit	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
rxpolplane	CHOICE	SAVE NOPRESET	EnumRange (H V)
locked	BOOLEAN	SAVE	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU Fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (ACU Warning)

9.49 Antenna-Pointing

The Antenna-Pointing logical device is a 'high level' driver for motor driven antennas. It is used together with a driver for the type of antenna controller used.

While the plain driver for the antenna controller simply supports to set the azimuth/elevation/pol-offset parameters of the antenna, the Antenna-Pointing logical device extends this with additional capabilities:

- It permits to set the antenna pointing by supplying a satellite orbit position. The Antenna-Pointing logical device computes the antenna pointing from the satellite position and sends these values to the low level antenna driver.
- It maintains a list of satellite names, stored together with the orbit position of each satellite and the (manually fine tuned) antenna pointing values for each satellite. The standard user interface for the Antenna-Pointing logical device contains a comfortable tool to handle this list.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing values, the selected satellite name and the orbit position of this satellite. You should use this page if you want to move the antenna to a certain orbit position.
- --- The list of satellite positions. You can select a satellite to move the antenna to with this page and you can maintain the satellite list.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Pointing device work properly.

parameter	***description***
config.reserved.lo.x	
config.reserved.lo.y	
config.reserved.hi.x	
config.reserved.hi.y	These settings give a hint to the RX-LineSettings-CDB devices, that a particular port of the antenna actually is reserved and not available for automatic antenna selection. These settings have no impact to the Antenna-Pointing device itself.
azId	The message ID controlling the antenna's azimuth parameter.

parameter	*description*
elId	The message ID controlling the antenna's elevation parameter.
polId	The message ID controlling the antenna's polarization parameter.
orbitId	The message ID controlling the antenna's orbit parameter.
polPlaneId	The message ID selecting the antenna's polarization plane. See section "Polarization plane handling" below.
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation.
beamwidth	The antenna's half 3dB beamwidth. A satellite is considered a pointed by the antenna, if the azimuth/elevation angles of the antenna match the stored pointing of a satellite within +/- beamwidth.

Operation modes

The Antenna-Pointing device knows three different modes of operation. This is to adapt the behaviour of the device to different types of antennas. By setting the azId, elId, polId and orbitId configuration parameter or leaving them empty you select the proper mode of operation for a particular antenna:

- **Normal 3-axis antenna** : For this antenna type, set the azId, elId and polId configuration parameters to the appropriate parameter identifiers. Leave the orbitId setting empty.In this operation mode, the device will control all three axes of the antenna and it will store complete (az/el/pol) pointing vectors in the satellite table. Only with this operation mode the device is capable to perform *orbit position* to *antenna pointing* calculations.
- **Polarmount type antenna** : For this antenna type, only set the azId configuration setting. Leave the other message ID parameters empty.The device will use the azimuth angle to control the antenna's orbit arc setting. This value together with a orbit position angle gets stored in the satellite table.
- **Intelligent mobile SNG antenna** : The third mode of operation is used with car mobile SNG antenna controllers which accept a satellite orbit position angle to point the antenna to this satellite. To enable this mode, leave the azId, elId and polId configuration settings empty and set the orbitId parameter to the appropriate value.The device will act as a simple satellite name to orbit position translator in this mode. Only the orbit position gets stored in the satellite table, the orbit position will be commanded to the antenna controller when a satellite is selected.

Polarization plane handling

The Antenna-Pointing device accepts an operational parameter called 'rxpolplane' (H/V). This

parameter may be used to select a certain receive polarization plane by turning the feed (This works with 3-axis antennas only.)

The device translates the commanded polarization angle when forwarding it to the antenna. If horizontal polarization is selected (which is the default if the parameter isn't used), the angle is forwarded as it is. With vertical polarization, the device either adds or subtracts 90 degrees before the value gets forwarded. The device decides if to add or to subtract based on the range of the polarization angle given by antenna controller driver.

If the configuration parameter polPlaneId is set, the device will forward the polarization plane setting to the given device / parameter rather than shifting the polarization angle. The destination may be a switch selecting the polarization plane or a comparable parameter of the antenna controller itself. This "pass-through" mode may be used with all types of antenna controllers, as long as there is a parameter to select the polarization plane.

Remarks

1. To make the antenna move to a satellite already stored in the list, double click to the satellite's entry in the - page.
2. Pressing the right mouse button on a satellite's entry in the list opens a context menu offering all funktions to manage the satellite list.
3. When entering a new satellite to the list, first move the antenna to it's calculated position by entering the satellite's orbit position at the - page. Optimize the antenna pointing. Then switch to the - page, add a new entry to the list and update this entry with the current antenna pointing.

Satellite position file

The Antenna-Pointing device stores the satellite positions in a text file in the base directory of the M&C/VLC software. It is named "XXXX.txt" where XXXX is the name of the Antenna-Pointing logical device. The file contains the satellite list as a table of semicolon separated fields. It may be imported by common spread sheet programs this way. The table columns are defined as follows.

1. satellite name
2. orbit position (deg. east)
3. antenna azimuth
4. antenna elevation
5. antenna polarization offset Be aware, that the numeric values are stored with a decimal dot (##.###), regardless of the default locale of your computer.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.reserved	CHOICE	SETUP	EnumRange (NO YES)
config.reserved.lo.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.lo.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.x	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.reserved.hi.y	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.azId	TEXT	SAVE SETUP	StringRange
config.elId	TEXT	SAVE SETUP	StringRange
config.polId	TEXT	SAVE SETUP	StringRange
config.orbitId	TEXT	SAVE SETUP	StringRange
config.polPlaneId	TEXT	SAVE SETUP	StringRange
config.beamwidth	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.satlist	GENERIC		ObjectRange generic object
config.longitude	FLOAT	SAVE SETUP	DoubleRange (-180.00 .. 180.00)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
satellite	CHOICE		EnumRange (?)
copositioned	TEXT	R/O	StringRange R/O
update	TEXT	NOPRESET	StringRange
azimuth	FLOAT	NOPRESET	DoubleRange DIS (0.0 .. 0.0)
elevation	FLOAT	NOPRESET	DoubleRange DIS (0.0 .. 0.0)
polarization	FLOAT	NOPRESET	DoubleRange DIS (0 .. 0)
orbit	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
rxpolplane	CHOICE	SAVE NOPRESET	EnumRange (H V)
polamount	BOOLEAN	R/O	BooleanRange R/O (true false)
fixed	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
orbitPassThrough	BOOLEAN	R/O	BooleanRange R/O (true false)
locked	BOOLEAN	SAVE	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU Fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (ACU Warning)

9.50 Antenna-Proxy

The Antenna-Proxy logical device translates parameters between different antenna controllers to the parameters from the Antenna-Pointing device.

This is useful if the antenna controller includes already a satellite list (like the SatService-ACU-ODM) but you need the standard interface of the Antenna-Pointing device for other logical devices like the RX-ChannelSelect devices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the selection state
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	*description*
satId	The message ID for the satellite name
azId	The message ID for the azimuth reading.
elId	The message ID for the elevation reading
polId	The message ID for the polarization reading

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.satId	TEXT	SAVE SETUP	StringRange
config.azId	TEXT	SAVE SETUP	StringRange
config.elId	TEXT	SAVE SETUP	StringRange
config.polId	TEXT	SAVE SETUP	StringRange
satellite	TEXT	SAVE	StringRange
azimuth	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
elevation	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
polarization	FLOAT	SAVE	DoubleRange (0 .. 0)

9.51 Antenna-Tracking

Please note, the Antenna-Tracking device is no longer maintained andshould not be used for new M&C installations.

The Antenna-Tracking logical device is a 'high level' driver for motor driven antennas. It is used together with a driver for the type of antenna controller used. To operate the device in step track modes, also a beacon receiver is required. The Antenna-Tracking device provides three mayor tracking modes:

- **STEP** --- In plain step track mode, the device performs small test steps with the antenna in regular intervals to see if the receive level becomes better or worse for another antenna position. With the knowledge of the antenna pattern's shape, the tracking engine computes an optimized position from measurements taken at the old position and at the new one. Many parameters such as the step size, averaging times and the overall repetition interval are configurable by the operator.
- **ADAPTIVE** --- The adaptive tracking mode works much like the plain step track, however

the tracking engine in background computes a mathematical model of the antenna motions from the step track results. After a tracking step the antenna is moved to the position computed from this model rather than to the measured maximum. This significantly smooths the path at which the antenna follows the satellite. Another great advantage of this mode is that the antenna follows the computed model even if the beacon reception fails. With the plain step track mode, the antenna would freeze in this case until the satellite beacon appears again.

- **PROGRAM** --- With the third tracking mode called program track, the antenna follows a position computed from the satellite's ephemeris data. The tracking engine accepts Keplerian elements in NASA 2-line format or the proprietary Intelsat ephemeris data format for this. For this tracking mode, no beacon reception is required, however the ephemeris data sets must be updated every few days.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Basic tracking parametersThe primary page contains the basic tracking parameters like the nominal antenna position or the mayor tracking mode. This page also gives access to the logging and analysis tools built into the tracking device. The most important parameters are explained below:**satellite name:** This ist the name of the satellite as it appears in the list of stored targets. When you read in ephemeris data which contains a satellite name, the latter gets copied into this field. You may however enter any other name.**nominal orbit:** With this parameter the satellite's nominal orbit position is specified. There are two applications for the nominal orbit value in the software: The tracking device is capable to evaluate the antenna pointing angles for the orbit position, you may move the antenna to the resulting position in order to find a new satellite quickly. Also the orbit value is used to identify and sort satellites in the target list.The nominal orbit position has to be entered as 'deg. east' To specify a satellite at 10° West, either enter -10.0 or 350.0, the software accepts both notations.**tracking mode:** With this parameter you select the major tracking mode. Available modes are:
  - STEP --- Plain step track mode.
  - ADAPTIVE --- Step track with mathematical model assistance / backup.
  - PROGRAM --- Program track following given ephemeris data.**interval:** Defines the overall tracking interval. The tracking device performs it's steps not faster than defined with this parameter. If an optimization step takes longer than this interval, this is not treated an a fault. The tracking device will start the next step immediately after the first one has finished. Set this value to 0 in order to force the tracking to operate at a maximum rate.The effective duration of one optimization step primarily depends on the integration times and delays defined with the advanced step track parameters. But also the fluctuation of the beacon level reading has an impact to the duration of a tracking step. If the standard variation of the readings will exceed a configurable threshold, the tracking algorithm will try to repeat the measurement several times.**az/el nominal:** These fields define the nominal position of the antenna for this satellite. The nominal position is the center

of the box in which the antenna tracks the satellite. The size of this box is set on the 'advanced tracking parameters' page described below. The nominal position also is that one the antenna moves to, if the satellite's tracking settings get recalled from a preset memory.Changing one of the nominal position values while the tracking mode is 'OFF' immediately moves the antenna to the given position. With tracking enabled, changing a nominal position value does not move the antenna, only the box limits are shifted accordingly.**polarization:** The polarization value directly controls the antenna's polarization offset motor (if available). None of the tracking algorithms changes this value, in must be optimized manually when a satellite is going to be tracked the first time.**beacon frequency:** This parameter controls the beacon receiver's receive frequency. The true RF receive frequency value is required, not an L-band frequency. It is necessary to enter this value even if the beacon receiver is not controlled by the tracking M&C system. This is because the tracking engine uses the beacon frequency to compute the tracking step sizes.**beacon pol.:** With beacon receivers which are capable to switch the input polarization, this parameter permits to select the polarization plane of the beacon to receive.**Y/T PLOT:** Clicking to the 'Y/T PLOT' button opens a window which shows a diagram of the antenna movement and the beacon level over the time. If you leave this window opened, it gets updated with each tracking step the antenna makes. See the paragraph 'built in analysis tools' below for further information.**X/Y PLOT:** Clicking to the 'X/Y PLOT' button opens a window which shows a diagram of the antenna movement in both axes. If you leave this window opened, it gets updated with each tracking step the antenna makes. See the paragraph 'built in analysis tools' below for further information.**STEP LOG:** Clicking to the 'STEP LOG' button opens a window where the program shows a live report what the tracking engine actually is doing. The log in this window is cleared with each new tracking step. If 'Y/T PLOT' diagram shows many unsuccessful tracking steps, watching the 'STEP LOG' reports may give an explanation why the tracking engine discards so many steps.**RESET POS:** Moves the antenna back to it's nominal position**RESET MEM:** Clears the tracking memory used by the ADAPTIVE tracking mode.**orbital model:** Displays the type and age of the orbital model used by the PROGRAM tracking mode. While in ADAPTIVE tracking mode 'ADAPTIVE' is displayed.**operation state:** Reports the actual operation state of the tracking engine. This mainly reflects the major tracking mode, but also contains some additional information, e.g. which type of ADAPTIVE model the tracking engine actually uses.**tracking step:** Reports the tracking step which recently has been done. This exactly matches the last line of the 'STEP LOG' display.**tracking offset:** These fields report the actual offset of the antenna pointing from the nominal position.**beacon level:** Displays the beacon level as actually read from the beacon receiver.

- --- Satellite listThe satellite list page permits to store tracking settings in a target memory pool which is unique to this instance of the Antenna-Tracking device. This is unlike the standard device preset function if the sat-nms software which stores device settings for all instances of one device type at a central place. The following functions are provided by this page:**Save actual satellite:** Clicking to this button stores the actual tracking parameters together with the nominal antenna pointing into a data file named after the actual satellite name followed by the nominal orbit position. If a file of this name

already exists, it gets replaced.**Delete seleted entry** Deletes the selected (yellow colored) target memory file.**Load selected parameters** Loads all tracking parameters except the nominal antenna pointing from the selected target memory. This is intended to be used to copy the tracking settings from an already known satellite to a new one, which actually is being set up.**Track selected satellite** Loads all settings including the nominal antenna pointing and the tracking mode. Moves the antenna to the nominal pointing position and starts the selected tracking mode there.

- --- Orbital modelAt this page you may define an orbital model for the satellite which is used with the PROGRAM tracking mode. The orbital model also may be used to find a satellite more precisely than by it's nominal orbital position.The tracking engine accepts either Keplerian elements in NASA 2-line format or Intelsat ephemeris data. The software automatically recognized the data format and chooses the appropriate orbital model. A common source for actual Keplerian ephemeris data is www.celetrak.com , weekly updated Intelsat orbital data may be found on the Intelsat web site.The easiest way to enter the orbital data to the tracking system is by copy & paste from the source web site. Clear the orbital data field, mark the desired orbital data in the web browser window and copy it into the orbital data entry field (use CTRL-C/CTRL-V on Windows based PC, the middle mouse button on Linux boxes like the sat-nms ACU indoor unit. For Intelsat data, copy the whole web page which shows the ephemeris data set. For Keplerian data copying the two lines containing the element data is required. When using data from Celestrak, you optionally may copy also the comment line above the element data, the satellite name will be adapted automatically in this case.Once you entered / copied the orbital data and change the input focus to another field, the software analyzed the data and updates the fields at the top of the page. The 'recognized orbital model' shows the type and age of the interpreted ephemeris data. The 'satellite name' and 'nominal orbit' are updated accordingly, if this information is present in the entered data. If you select the PROGRAM tracking mode on the primary parameter page, the antenna follows the position of the satellite as computed from the ephemeris data.Using the buttons below the orbital data entry field you may compute the actual antenna pointing either from the entered ephememeris data or from the satellite's nominal orbit position. Clicking to the large button at the bottom move the antenna to the computed position. This function is intended to be used to find a new satellite quickly. If there is no orbital data available for the satellite you search, you may enter it's nominal orbit position directly and compute the antenna pointing from this.The accuracy of the calculated antenna positions heavily depends on the antenna pointing calibration and on the accuracy of the antenna's geodetic location as defined with the configuration parameters.

- --- Advanced step track parametersThis page contains a number of parameter which are used to tune the tracking algorithm. Most of these parameters affect the STEP and ADAPTIVE tracking modes as well.**step size:** This parameter controls the sizes of the search steps the tracking engine performs in the STEP and ADAPTIVE modes. With the step size set to 100%, the elevation test step is one half of the antenna's 1dB bandwidth. The azimuth step size is computed the same way, but divided by *cos(elevation)* to deskew look angle to the satellite. The antenna's beamwidth (approximately) is computed from the beacon frequency and the antenna diameter. A good initial value for the step size is 60%.**settling time:** The settling time is the time the tracking engine

waits after it issued an antenna pointing command until it takes the first sample of the beacon level. Specially with small steps of the antenna position, the tracking engine has no real feedback when the antenna has reached it's final position. This delay time is used to ensure that the antenna pointing command has been executes and the beacon level has settled.**step mode:** In STEP mode, this parameter controls how the tracking engine selects the direction of a search step. With the SLOW step mode, search steps alternately are made in the one or the other direction. This step mode gives good results with satellites that move only a small amount. With the FAST step mode, each search step is made in the same direction of the last successful optimization step. With any unsuccessful step the direction changes. This mode is recommended to track inclined orbit satellites, because the tracking does not perform search steps against the satellite movement in the majority of cases. In the ADAPTIVE tracking mode, the tracking engine observes the step mode selection only as long as it cannot compute a usable model from the records steptrack data. Later it uses the the modeled antenna pointing to decide where to go for a search step.**max optimization:** The step track algorithm uses the measured beacon levels before a search step and after it together with some knowledge about the shape of the antenna beam to compute the peak location. Unexpected variations of the beacon level may disturb the measurement and may result to a wrong peak evaluation. To minimize the impact of this to the antenna pointing, the tracking algorithm limits the amount of an optimization step to this value. Setting this parameter to 0% forces the tracking engine to limit the peak position to a window between the antenna positions before and after the search step. Increasing the 'max optimization' parameter widens this window at both edges with the actual search step size multiplied by the 'max optimization' value.With the ADAPTIVE tracking mode this parameter is used, too. Here however it acts as a fault threshold: If a peak is outside, is is marked as invalid and not included to the tracking model computation.**az/el search step:** These fields show the actual search step sizes as absolute angles for you orientation. The fields get updated each time you change the 'step size' parameter.**az/el window:** These parameters define a rectangular window around the nominal antenna position. The tracking engine limits the optimization movements to this window.**beacon averaging time:** This parameter defines the time, while samples read from the beacon receiver get averaged to achieve one beacon level value. Recommended values are between 15 and 30 seconds, if the beacon receiver itself performs an effective averaging/smoothing of the level reading, shorter values may be possible.**max standard deviation:** While the program averages the beacon level readings, it also computes the standard deviation of the reading as a measure of the (short term) stability of readings. If the standard deviation exceeds this limit, the measurement is discarded.**max signal drift:** Before performing a search step, the tracking algorithm requires two beacon measurement series to differ not more than the value defined here. Otherwise the beacon level is assumed to drift, the search step is postponed or canceled.**max drop in search step:** If with a search step the beacon level drops more than the value set here, the antenna moves back to the previous or predicted position, no peak evaluation is done.**min beacon level:** If the measured beacon level falls below this value, a beacon level fault is risen and the measurement is marked as invalid. With the STEP tracking mode no tracking is performed until the beacon level is above the threshold again. In ADAPTIVE mode, the tracking engine tracks the satellite following the mathematical model made from the step track peaks.

- --- Faults and fault mask**Beacon receiver:** The beacon receiver device states a fault. This may be a hardware fault or a loss of signal. It is recommended to switch of any level threshold monitoring at the beacon receiver and let the tracking device monitor the beacon level. The tracking device applies threshold monitoring only during the measurement phases, signal dropouts during the antenna movement phases are ignored.**Beacon level:** The measured beacon level is below the threshold set at the advanced step track parameters page.**AZ/EL window limit:** The recent step track peak found is outside the window limits set at the advanced step track parameters page. The antenna has been moved to the window limit rather than to the peak position. The fault stays active until the next tracking step.**Antenna:** The antenna positioning controller states a fault. This may be a motor failure, limit switch or another specific antenna fault.**Search step size:** The tracking engine is not capable to compute a valid tracking step size from the given data. This happen if either the antenna diameter (a setup parameter) or the beacon frequency is not set properly. The fault stays active until the missing value is set.**AZ/EL Model match:** In ADAPTIVE tracking mode, a tracking step found a peak which considered to be invalid at it is too far away from the position computed by the model. The fault stays active until the next tracking step.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the Antenna-Tracking device work properly.

parameter	***description***
config.azimuthId	Enter here the parameter ID of the azimuth target value the tracking unit shall use. Example: 'ODU.azimuth'
config.stateAzId	Enter here the parameter ID of the azimuth read out value the tracking unit shall use. Example: 'ODU.state.az'
config.elevationId	Enter here the parameter ID of the elevation target value the tracking unit shall use. Example: 'ODU.elevation'
config.stateElId	Enter here the parameter ID of the elevation read out value the tracking unit shall use. Example: 'ODU.state.el'
config.polarizationId	Enter here the parameter ID of the polarization offset target value the tracking unit shall use. The tracking unit does no polarization, however a fixed polarization offset value is stored together with each target. Example: 'ODU.polarization'
config.antennaFaultId	Enter here the parameter ID of the antenna fault, the tracking unit shall monitor.

parameter	***description***
config.bcrxFreqId	Enter here the parameter ID of the beacon receiver frequency setting. The tracking unit sends the receive frequency stored with a target to this parameter (RF frequency, not L-band). Leave this field empty, if the beacon receiver frequency cannot be controlled through the sat-nms software. Please note, that in the tracking unit the beacon frequency must be set to the correct value even if it is not passed to the beacon receiver. This is because the tracing unit computes the step track search sizes from the beacon frequency.
config.bcrxLevelId	Enter here the parameter ID of the beacon receiver level reading. The value is expected to be a receive level in dBm or a C/N value [dB]. To read the analog output value of a tracking receiver, you need an external device which reads this voltage and converts the readint to dBm. The sat-nms Antenna Control Unit ODM is capable to do this.
config.bcrxFTrackId	If you are using a sat-nms L-band beacon receiver together with this tracking device, enter 'BCRX.ftrack.fast' here, replace BCRX by the device name of the beacon receiver. This tells the tracking engine to perform a synchronous frequency tracking step on the beacon receiver before each antenna tracking step.
config.bcrxFaultId	Enter here the parameter ID of the beacon receiver fault, the tracking unit shall monitor.
config.xDiameter	This parameter tells the tracking device the width of the antenna dish. The tracking device computes the azimuth tracking step size from this parameter and the beacon frequency value.
config.yDiameter	This parameter tells the tracking device the height of the antenna dish. The tracking device computes the elevation tracking step size from this parameter and the beacon frequency value.
longitude	The antenna's longitude (degree east). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation
latitude	The antenna's latitude (degree north). Provide this value with an accuracy of 0.01° to ensure a precise orbit to pointing calculation
altitude	The antenna's altitude over sea. Provide this value with an accuracy of 10 meters to ensure a precise orbit to pointing calculation

Built in analysis tools

The Antenna-Tracking device is capable to show two different types of diagrams which support

you to monitor and optimize the tracking behaviour.

The first diagram tool is the X/Y display of the antenna movements. This display works completely autoscaled, it shows one green dot for each antenna position the tracking memorizes (usually the recent 48 hours). The actual antenna pointing is marked with a slightly larger yellow dot. After at least 24 hours of continuous operation, the antenna position dots give the curve of the daily movement of the satellite, from the antenna's angle of look. With a mouse click to the diagram area, the display cas be switched into a Y/T diagram showing the recent 48 hours. This condensed display of the azimuth, elevation and beacon level variations over 48 hours is meant as an overview display. A second mouse click switches back to the previous view.

The separate Y/T display window gives more detailed information about what the tracking engine did during the recorded 48 hours. It provides variable scales, and permits to display additional information. The display shows either the azimuth or the elevation pointing found at the beginning of each each tracking step together with the beacon level measured at this time. The following additional information may be included to the diagram:

- --- If this toolbar button is pressed, the diagram shows the search step and the beacon level change resulting from this as a dimmed peak at the left edge of each data point. You can easily monitor the signal degradation which is caused temporarily by the tracking search steps with this function enabled.
- --- If this toolbar button is pressed, the diagram includes the evaluated peak positions. In STEP track mode, this actually is the location where the antenna gets moved to. In ADAPTIVE tracking mode, the antenna moves to the location resulting from the mathematical model which is computed from the step track peaks. In the latter case, the peak positions may be interpreted as the actual difference between the modeled pointing and the peak position.

Recommended beacon receiver settings

Below some recommended settings for the sat-nms L-band beacon receiver for the use with the tracking device are listed. These settings have proved to work satisfactory in most cases. If you are using a third party beacon receiver, use analogous values a starting point.

- Measurement bandwidth --- 30kHz
- Post detector filter --- 0.5 Hz
- Frequency tracking --- OFF (the tracking device switches this ON/OFF automatically with a sat-nms L-band beacon receiver)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.azimuthId	TEXT	SAVE SETUP	StringRange
config.stateAzId	TEXT	SAVE SETUP	StringRange
config.elevationId	TEXT	SAVE SETUP	StringRange
config.stateElId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.antennaFaultId	TEXT	SAVE SETUP	StringRange
config.bcrxFreqId	TEXT	SAVE SETUP	StringRange
config.bcrxLevelId	TEXT	SAVE SETUP	StringRange
config.bcrxPolId	TEXT	SAVE SETUP	StringRange
config.bcrxInhibitId	TEXT	SAVE SETUP	StringRange
config.bcrxFaultId	TEXT	SAVE SETUP	StringRange
config.xDiameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.yDiameter	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)

name	*type*	*flags*	*range*
azimuth	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
elevation	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
polarization	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
trackMode	CHOICE	SAVE	EnumRange (OFF STEP ADAPTIVE PROGRAM)
polPlane	CHOICE		EnumRange DIS (H V)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
interval	INTEGER	SAVE	IntegerRange (0 .. 60)
name	TEXT	SAVE	StringRange
elements	TEXT	SAVE	StringRange
orbit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
stepSize	INTEGER	SAVE	IntegerRange (0 .. 200)
stepMode	CHOICE	SAVE	EnumRange (FAST SLOW)
optimizeSize	INTEGER	SAVE	IntegerRange (0 .. 200)
averagingTime	INTEGER	SAVE	IntegerRange (1 .. 120)
settlingTime	INTEGER	SAVE	IntegerRange (1 .. 120)
dazMax	FLOAT	SAVE	DoubleRange (0.0 .. 10.0)
delMax	FLOAT	SAVE	DoubleRange (0.0 .. 10.0)
driftMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
dropMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
sdevMax	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
beaconThreshold	FLOAT	SAVE	DoubleRange (-100.0 .. 0.0)
state.saz	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.sel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.daz	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.del	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.mode4	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.elements	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
evalByOrbit	TEXT	NOPRESET	StringRange
evalByElements	TEXT	NOPRESET	StringRange
moveEvaluated	TEXT	NOPRESET	StringRange
slist.select	CHOICE	NOPRESET	EnumRange ()
slist.action	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ window limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL window limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Antenna)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Search step size)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Model match)
faults.08	ALARM	R/O	AlarmFlagRange R/O (EL Model match)

9.52 Antenna-Wind-Protection

The Antenna-Wind-Protection logical device protects an antenna from high wind speeds by either stopping all antenna movement and fastening the brakes or by moving the antenna elevation up to a stow position where the wind load to the antenna's construction is minimized.

The device monitors two wind speeds, each as a momentary and an averaged value. The wind

speed measurement values are typically provided by a Dual-Wind-Speed logical device, but other sources are configurable. It takes the maximum value from the two wind speed meters individually for the momentary and the average reading. These max values are compared to configurable limits in order to evaluate an action to perform. The device provides two protection mechanisms, "stop" and "stow", which are working independently from each other.

Antenna Stop procedure

The device compares the momentary and the averaged wind speed to a pair of limit values. If either the momentary or the averaged value exceeds its limit, the antenna gets stopped by execution of the "stop" macro. This macro should contain all antenna settings necessary to lock the antenna safely in its actual position (switch off tracking, fasten brakes).

Antenna Stow procedure

In parallel to the procedure described above, the device compares the momentary and the averaged wind speed to another pair of limit values. If either the momentary or the averaged value exceeds its limit, the "stow" procedure gets triggered. The stow procedure consists of the following steps:

1. The device issues a warning and wait for a configurable time (default is 30 seconds). During this time the procedure can be aborted without manipulating the antenna's state.
2. After the initial delay has elapsed, the device executes the "pre-stow" macro. This macro should contain any antenna settings which are necessary to prepare the antenna to be moved to its stow position.
3. After the execution of the pre-stow macro has finished, the device commands the antenna to move the elevation to its stow position.
4. Now the antenna waits until the elevation reading matches the commanded stow angle within 0.05 degrees.
5. When the stow position is reached, the devices executes the "post-stow" macro. This macro should contain all antenna settings which are required to lock the antenna in the stow position.
6. After the macro execution has finished, the device turns into the STOWED state.

The procedure may be aborted at any time by clicking the "ABORT" button, but this does not stop running macros nor does it stop the elevation to move to the stow position if this already has been commanded. Clicking "ABORT" simply turns the device into the ABORTED state.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows and controls the state of the protection procedures.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Antenna-Wind-Protection device.

parameter	***description***
config.speed.1.act.id	The parameter ID of the wind speed 1 actual / momentary value.
config.speed.2.act.id	The parameter ID of the wind speed 2 actual / momentary value.
config.speed.1.avg.id	The parameter ID of the wind speed 1 average value.
config.speed.2.avg.id	The parameter ID of the wind speed 2 average value.
config.elevation.state.id	The parameter ID of the antenna's elevation angle reading.
config.elevation.cmd.id	The parameter ID of the antenna's elevation target value setting.
config.elevation.stow	The elevation angle to command when stowing the antenna.
config.stopMacro	The name of the macro to execute in order to stop the antenna.
config.preStowMacro	The name of the macro to execute before stowing the antenna.
config.postStowMacro	The name of the macro to execute after stowing the antenna.
config.act.stopLimit	If the higher value of the actual wind speeds exceeds this value, the antenna gets stopped.
config.act.stowLimit	If the higher value of the actual wind speeds exceeds this value, the antenna gets stowed.
config.avg.stopLimit	If the higher value of the averaged wind speeds exceeds this value, the antenna gets stopped.
config.avg.stowLimit	If the higher value of the averaged wind speeds exceeds this value, the antenna gets stowed.
config.stowDelay	The time to wait until the stowing procedure actually starts (seconds).

Setting up the device

The Antenna-Wind-Protection device depends on a number of macros you have to supply in order to make the device work properly. Macros have been chosen for this because each type of antenna controller requires different settings and commands to perform the actions before and after the stow procedure or to stop the antenna safely.

The device suggests names for these macros which contain the name of the Antenna-Wind-Protection device and a suffix describing what the macro shall do. You may use these names or names of your choice. However, all three macros must exist with the names set at the device's configuration parameters or the device will signal a fault.

Manual release procedure

If the device stopped or stowed the antenna due to high wind speed, the operator has to restore the antenna's state and the state of the Antenna-Wind-Protection device to normal operation conditions. The procedure ist almost the same for stop and stow conditions. It also applies when an antenna stow has been aborted and the pre-stow macro already has been executed.

1. Wait until the wind has calmed down to speeds the antenna mount han handle.
2. Point the antenna back to the satellite, activate tracking (if applicable).
3. Finally click "RELEASE" at the Antenna-Wind-Protection device to re-activate the wind protection procedures.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.speed.1.act.id	TEXT	SAVE SETUP	StringRange
config.speed.2.act.id	TEXT	SAVE SETUP	StringRange
config.speed.1.avg.id	TEXT	SAVE SETUP	StringRange
config.speed.2.avg.id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.elevation.state.id	TEXT	SAVE SETUP	StringRange
config.elevation.cmd.id	TEXT	SAVE SETUP	StringRange
config.elevation.stow	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.stopMacro	TEXT	SAVE SETUP	StringRange
config.preStowMacro	TEXT	SAVE SETUP	StringRange
config.postStowMacro	TEXT	SAVE SETUP	StringRange
config.act.stopLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.act.stowLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.avg.stopLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.avg.stowLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.stowDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 120)
speed.act	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed.avg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
stop.enable	CHOICE	SAVE	EnumRange (DISABLED ENABLED)
stop.force	TEXT	NOPRESET	StringRange
stop.release	TEXT	NOPRESET	StringRange
stop.state	TEXT	R/O	StringRange R/O
stop.overAvgLimit	BOOLEAN	R/O	BooleanRange R/O (true false)
stop.overActLimit	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
stow.enable	CHOICE	SAVE	EnumRange (DISABLED ENABLED)
stow.force	TEXT	NOPRESET	StringRange
stow.release	TEXT	NOPRESET	StringRange
stow.abort	TEXT	NOPRESET	StringRange
stow.state	TEXT	R/O	StringRange R/O
stow.overAvgLimit	BOOLEAN	R/O	BooleanRange R/O (true false)
stow.overActLimit	BOOLEAN	R/O	BooleanRange R/O (true false)
stow.delayCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stop.force.en	BOOLEAN	R/O	BooleanRange R/O (true false)
stop.release.en	BOOLEAN	R/O	BooleanRange R/O (true false)
stow.force.en	BOOLEAN	R/O	BooleanRange R/O (true false)
stow.release.en	BOOLEAN	R/O	BooleanRange R/O (true false)
stow.abort.en	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Stop macro)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Pre stow macro)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Post stow macro)

9.53 AntwerpSpace-G3-Demod

Device driver for the 3rd generation AntwerpSpace Omnisat G3 Demodulator (IPC based solution).

It can be used to control one demodulator instance in a Ominsat G3. If more than one demodulator is used inside a chassis, additional instances of this driver can be added under the same device thread. Assigning the dedicated demodulator can be done via device setup *config.boardId* . Additionally mixed board combinations can be covered by adding an Modulator HDRM AntwerpSpace-G3-Mod to the same thread/chassis.

This driver interfaces via REST HTTP to the demodulator web server it requires expert privileges to be able to do a counter reset / carrier reaquisition. Beside that the driver only contains HDRD operational access parameters. It does not contain parameters for: CCSDS-FEP (configurational) and DVBS-FEP (configurational and monitoring).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The HDRD monitoring parameters
- --- The HDRD configuration parameters
- --- The CCSDS-FEP monitoring parameters
- --- Load units mission/configuration presets
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The REST HTTP specification for the device. The full address specification consists of the following fields: *aaa.bbb.ccc.ddd:pppp username password* *aaa.bbb.ccc.ddd* The IP address of the device to control in 'dotted quad' notation. This field is mandatory. *pppp* The IP port number where the device accepts http requests. The port number may be omitted, the driver uses the standard port 80 in this case. **username password* The Web access username and password for expert access: (Default: **og3expert** and **og3expertpwd**)
boardId	Defines the board index where the HDRD demodulator is installed.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.temp.adc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.board	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.fpgaInt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.fpgaExt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.demod.load	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.chan.load	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.arm.load	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.uuid	TEXT	R/O	StringRange R/O
internal.reply	TEXT	R/O	StringRange R/O
internal.mission	TEXT	R/O	StringRange R/O
internal.configuration	TEXT	R/O	StringRange R/O
config.boardId	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
rx.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rfBw	INTEGER	R/O	IntegerRange R/O (0 .. 10000)

name	*type*	*flags*	*range*
meas.symbolrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.bitRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.q.bitRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.q.symbolrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
q.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
acqTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reacqCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.q.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
perturbation	FLOAT	R/O	DoubleRange R/O (0.00000000000 .. 0.00000000000)
jitter	FLOAT	R/O	DoubleRange R/O (0.0000000000000 .. 0.0000000000000)
prbs1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
prbs1.resync	INTEGER	R/O	IntegerRange R/O (0 .. 0)
prbs1.polarity	CHOICE	R/O	EnumRange R/O (NORMAL INVERTED)
prbs2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
prbs2.resync	INTEGER	R/O	IntegerRange R/O (0 .. 0)
prbs2.polarity	CHOICE	R/O	EnumRange R/O (NORMAL INVERTED)

name	*type*	*flags*	*range*
perturbationDll1	FLOAT	R/O	DoubleRange R/O (0.00000000000 .. 0.00000000000)
perturbationDll2	FLOAT	R/O	DoubleRange R/O (0.00000000000 .. 0.00000000000)
jitterDll1	FLOAT	R/O	DoubleRange R/O (0.0000000000000 .. 0.0000000000000)
jitterDll2	FLOAT	R/O	DoubleRange R/O (0.0000000000000 .. 0.0000000000000)
meas.convDec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.convDec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
prbs.convDec1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
prbs.convDec2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tcmDec	FLOAT	R/O	DoubleRange R/O (0.0000000000000 .. 0.0000000000000)
reference	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL)
flags.carrier	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.agc	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.perturbation	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.jitter	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.syncI	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.syncQ	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.jitterDll1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.jitterDll2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.perturbationDll1	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
flags.perturbationDll2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.convDec1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.convDec2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.tcmDec	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.reference	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ccsds	BOOLEAN	R/O	BooleanRange R/O (true false)
ccsds.bitSlipFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.syncWordErr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.asmBer	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ccsds.fs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.fsCheck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.fsLock	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.fsFly	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.fsToSearch	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsCorFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsUncorFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsUncorBlock	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsCorHdr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsUncorHdr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsCorCodeHdr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.rsCorCodeData	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.crcErr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.ldpcFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.ldpcCorFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.ldpcUncorFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ccsds.ldpcDecoder	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccsds.ldpcDisambiguation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
adcRate	INTEGER		IntegerRange (0 .. 2500)
iBitRate	FLOAT	R/O	DoubleRange R/O (1.0000 .. 975.0000)
qBitRate	FLOAT	R/O	DoubleRange R/O (1.0000 .. 975.0000)
iSymbolRate	FLOAT		DoubleRange (1.0000 .. 325.0000)
qSymbolRate	FLOAT		DoubleRange (1.0000 .. 325.0000)
modulation	CHOICE		EnumRange (BPSK QPSK SQPSK UQPSK AQPSK 8PSK GMSK)
inpSelect	CHOICE		EnumRange (INTERNAL EXTERNAL)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rfFrequency	FLOAT		DoubleRange (2000.000 .. 20000.000)
loPolarity	CHOICE		EnumRange (NORMAL INVERTED)
pll	CHOICE		EnumRange (OPEN CLOSED)
searchRange	INTEGER		IntegerRange (0 .. 3000)
preLockBw	INTEGER		IntegerRange (0 .. 100)
postLockBw	INTEGER		IntegerRange (0 .. 100)
damping	INTEGER		IntegerRange (0 .. 2)
chanFilter	CHOICE		EnumRange (RECTANGULAR SRRC RC)
chanFilterRolloff	FLOAT		DoubleRange (0.10 .. 1.00)
diffDec1	CHOICE		EnumRange (OFF ON)
diffDec2	CHOICE		EnumRange (OFF ON)
convDec1	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
convDec2	CHOICE		EnumRange (OFF ON)
diffDecMode1	CHOICE		EnumRange (QPSK 0 QPSK 1 QPSK 2 QPSK 3 QPSK 4 QPSK 5 QPSK 6 QPSK 7 QPSK 8 BPSK NRZ-M BPSK NRZ-S)
diffDecMode2	CHOICE		EnumRange (QPSK 0 QPSK 1 QPSK 2 QPSK 3 QPSK 4 QPSK 5 QPSK 6 QPSK 7 QPSK 8 BPSK NRZ-M BPSK NRZ-S)
convDecMode	CHOICE		EnumRange (SERIAL PARALLEL)
convDec1Rate	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8)
convDec2Rate	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8)
prbsType1	CHOICE		EnumRange (INTERVAL ACCUMULATED)
prbsType2	CHOICE		EnumRange (INTERVAL ACCUMULATED)
ber1DataSelect	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
ber2DataSelect	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
eclDataA	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
eclDataB	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)

name	*type*	*flags*	*range*
ethDataA	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
ethDataB	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
recDataA	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
recDataB	CHOICE		EnumRange (DEMOD-I DEMOD-Q INNER-CH1 INNER-CH2 CCDS-CH1 CCDS-CH2 DVBS-FEP 8PSK-TCM DEMOD-MERGE INNER-MERGE)
prbsPoly1	CHOICE		EnumRange (PRBS 7 PRBS 10 PRBS 11 PRBS 15 PRBS 23 PRBS 31)
prbsPoly2	CHOICE		EnumRange (PRBS 7 PRBS 10 PRBS 11 PRBS 15 PRBS 23 PRBS 31)
prbsResync1	CHOICE		EnumRange (45% 25% 10% 5%)
prbsResync2	CHOICE		EnumRange (45% 25% 10% 5%)
clockEdge	CHOICE		EnumRange (RISING FALLING)
clockEcl	CHOICE		EnumRange (OFF ON)
demodAmbiguity	CHOICE		EnumRange (OFF ON)
dataInv	CHOICE		EnumRange (OFF ON)
dataInvQ	CHOICE		EnumRange (OFF ON)
dataInvDec1	CHOICE		EnumRange (OFF ON)
dataInvCc1	CHOICE		EnumRange (OFF ON)
bitOrderRaw	CHOICE		EnumRange (ODD EVEN)
bitOrderInner	CHOICE		EnumRange (ODD EVEN)

name	*type*	*flags*	*range*
mission	CHOICE		EnumRange (?)
configuration	CHOICE		EnumRange (?)
loadConfiguration	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
recordstop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Carrier Loop)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Timing Loop)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AGC Loop)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PLL Perturbation)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PLL Jitter)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Timing Loop I)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Timing Loop Q)
faults.08	ALARM	R/O	AlarmFlagRange R/O (DLL1 Perturbation)
faults.09	ALARM	R/O	AlarmFlagRange R/O (DLL1 Jitter)
faults.10	ALARM	R/O	AlarmFlagRange R/O (DLL2 Perturbation)
faults.11	ALARM	R/O	AlarmFlagRange R/O (DLL2 Jitter)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Conv Dec1)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Conv Dec2)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TCM Dec)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Reference)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.17	ALARM	R/O	AlarmFlagRange R/O (CCSDS FEP Sync)

9.54 AntwerpSpace-G3-Mod

Device driver for the 3rd generation AntwerpSpace Omnisat G3 Modulator (IPC based solution).

It can be used to control one modulator instance in a Ominsat G3. If more than one modulator is used inside a chassis, additional instances of this driver can be added under the same device thread. Assigning the dedicated modulator can be done via device setup *config.boardId* . Additionally mixed board combinations can be covered by adding AntwerpSpace-G3-Demod) to the same thread/chassis.

This driver interfaces via REST HTTP to the modulator web server it requires only operator privileges.The driver only contains HDRM operational access parameters. It dows not contain parameters for: CCDS-FEP and DVBS-FEP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The HDRM monitoring parameters
- --- The HDRM configuration parameters
- --- Load units mission/configuration presets
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*

parameter	*description*
address	The REST HTTP specification for the device. The full address specification consists of the following fields: *aaa.bbb.ccc.ddd:pppp username password* *aaa.bbb.ccc.ddd* The IP address of the device to control in 'dotted quad' notation. This field is mandatory. *pppp* The IP port number where the device accepts http requests. The port number may be omitted, the driver uses the standard port 80 in this case. *username password* The Web access username for operational access: (Default: **operator**)
boardId	Defines the board index where the HDRD demodulator is installed.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.tempFpga	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.reply	TEXT	R/O	StringRange R/O
internal.mission	TEXT	R/O	StringRange R/O
internal.configuration	TEXT	R/O	StringRange R/O
config.boardId	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
flags.sync	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.dataRateA	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.dataRateB	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.reference	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL)
meas.adcLevel2	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.signalGain1	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.signalGain2	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.noiseGain1	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.noiseGain2	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.adcLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fifoLevel1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fifoLevel2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iSymbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
meas.qSymbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.iDataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.qDataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE		EnumRange (BPSK QPSK SQPSK UQPSK AQPSK 8PSK GMSK)
input	CHOICE		EnumRange (INTERNAL-PRBS DIRECT-FROM-NETWORK REPLAY-FROM-DISK)
data1	CHOICE		EnumRange (DATA-A DATA-B)
data2	CHOICE		EnumRange (DATA-A DATA-B)
tx.mod	CHOICE		EnumRange (ON OFF)
tx.on	CHOICE		EnumRange (ON OFF)
tx.power	FLOAT		DoubleRange (-50.0 .. 0.0)
iBitRate	FLOAT	R/O	DoubleRange R/O (1.0000 .. 975.0000)
qBitRate	FLOAT	R/O	DoubleRange R/O (1.0000 .. 975.0000)
iSymbolRate	FLOAT		DoubleRange (1.0000 .. 325.0000)
qSymbolRate	FLOAT		DoubleRange (1.0000 .. 325.0000)
sweepRate	FLOAT		DoubleRange (0.0 .. 100.0)
sweepDeviation	FLOAT		DoubleRange (0.0 .. 3000.0)
noise	CHOICE		EnumRange (ON OFF)
esNo	FLOAT		DoubleRange (-10.0 .. 100.0)
signalOff	CHOICE		EnumRange (ON OFF)
noiseOff	CHOICE		EnumRange (ON OFF)
loPolarity	CHOICE		EnumRange (NORMAL INVERTED)

name	*type*	*flags*	*range*
prbsPoly1	CHOICE		EnumRange (PRBS 7 PRBS 10 PRBS 11 PRBS 15 PRBS 23 PRBS 31)
prbsPoly2	CHOICE		EnumRange (PRBS 7 PRBS 10 PRBS 11 PRBS 15 PRBS 23 PRBS 31)
chanFilter	CHOICE		EnumRange (RECTANGULAR SRRC)
chanFilterRolloff	INTEGER		IntegerRange (10 .. 100)
chanEnc	CHOICE		EnumRange (4D-TCM DIFFERENTIAL BYPASS)
chanEncEff	CHOICE		EnumRange (2.00 2.25 2.50 2.75)
diffEnc1	CHOICE		EnumRange (OFF ON)
diffEnc2	CHOICE		EnumRange (OFF ON)
convEnc1	CHOICE		EnumRange (OFF ON)
convEnc2	CHOICE		EnumRange (OFF ON)
diffEncMode1	CHOICE		EnumRange (QPSK 0 QPSK 1 QPSK 2 QPSK 3 QPSK 4 QPSK 5 QPSK 6 QPSK 7 QPSK 8 BPSK NRZ-M BPSK NRZ-S)
diffEncMode2	CHOICE		EnumRange (QPSK 0 QPSK 1 QPSK 2 QPSK 3 QPSK 4 QPSK 5 QPSK 6 QPSK 7 QPSK 8 BPSK NRZ-M BPSK NRZ-S)
convEncMode	CHOICE		EnumRange (SERIAL PARALLEL)
convEncRate1	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8)
convEncRate2	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8)
dataMerge	CHOICE		EnumRange (SPLIT MERGE)
fepMode1	CHOICE		EnumRange (UNSYNC DVBS-FEP CCSDS-FEP)
fepMode2	CHOICE		EnumRange (UNSYNC DVBS-FEP CCSDS-FEP)
imbalance	INTEGER		IntegerRange (0 .. 10)
sqpskDelay	CHOICE		EnumRange (DELAY-I DELAY-Q)
mission	CHOICE		EnumRange (?)

name	*type*	*flags*	*range*
configuration	CHOICE		EnumRange (?)
loadConfiguration	TEXT	NOPRESET	StringRange
replaystart	TEXT	NOPRESET	StringRange
replaystop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.02	ALARM	R/O	AlarmFlagRange R/O (DAC Sync)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External Ref)

9.55 AntwerpSpace-IBB-Modem

This page describes the *Antwerpspace-IBB-Modem* device driver and its device window. The *Antwerpspace-IBB-Modem* device driver uses the CCCOM protocol and reads the following tables: version and system table, the modulator Table (component #40), the subcarrier modulator table (component #30), the suppressed carrier modulator table #1 (component #36), the demodulator table (component #4), the psk subcarrier demodulator table (component #8), suppressed carrier demodulator table (component #12), the telemetry table (component #16). Beside that it allows setting a small selection of parameters from this tables, start and stop of the modules

It was developed base on ICD v4.07 from 2021/12/15

The IBB requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 5000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
The default Monitoring Port is 5000 It allows up to 5 connections at a time.
The default Command Port is 5001. It allows only one connection at a time.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Modulator settings and status
- --- ^Demodulator 1 settings and status
- --- Telemetry settings and status
- --- Baseband Session and Diversity calibration
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	No address required
cmdPort	Defines the command port. Standard port is 5001
mod.defaultState	Defines the the default modulation state when loading a new configuration (ON/OFF)
mod.defaultLoop	Deines the default IF loop state (INTERNAL/EXTERNAL). If set to internal loop, the modulator output is directly connected (at IF level) to all the inputs.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.reference	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.mod.defaultState	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.mod.defaultLoop	CHOICE	SETUP	EnumRange (EXTERNAL INTERNAL)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.cf	HEX	R/O	HexRange R/O (0 .. 0)
internal.fn	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy1	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy2	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy3	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy4	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy5	HEX	R/O	HexRange R/O (0 .. 0)
internal.sy6	HEX	R/O	HexRange R/O (0 .. 0)
internal.cmd	TEXT	R/O	StringRange R/O
mod.1.type	CHOICE	R/O	EnumRange R/O (PM FM)
mod.1.frequency	FLOAT		DoubleRange (65.000000 .. 75.000000)
mod.1.level	FLOAT		DoubleRange (-30.0 .. 0.0)
mod.1.sweepCycles	INTEGER	R/O	IntegerRange R/O (1 .. 20)
mod.1.sweepLimit1	FLOAT	R/O	DoubleRange R/O (-500.000 .. 500.000)
mod.1.sweepLimit2	FLOAT	R/O	DoubleRange R/O (-500.000 .. 500.000)
mod.1.sweepRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.configFile	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
mod.1.state	TEXT	R/O	StringRange R/O
mod.1.carrier	CHOICE		EnumRange (OFF ON)
mod.1.modulation	CHOICE		EnumRange (OFF ON)
mod.1.ifLoop	CHOICE	R/O	EnumRange R/O (EXTERNAL INTERNAL)
mod.1.sweep	TEXT	R/O	StringRange R/O
mod.1.video	CHOICE		EnumRange (OFF ON)
mod.1.sweepcmd	INTEGER	NOPRESET	IntegerRange (0 .. 2)
mod.1.sub.priority	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.pmIndex	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
mod.1.sub.fmDeviation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.waveform	CHOICE	R/O	EnumRange R/O (SINE SQUARE)
mod.1.sub.symbolRate	INTEGER		IntegerRange (0 .. 0)
mod.1.sub.sendPreamble	CHOICE	R/O	EnumRange R/O (NO PLOP-1 PLOP-2)
mod.1.sub.preambleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.minIdleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.maxIdleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sub.inversion	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
mod.1.sub.differential	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sub.convolutional	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sub.g2OutInversion	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sub.mirrorG1G2	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sub.format	CHOICE	R/O	EnumRange R/O (NRZ SPL)
mod.1.sub.configFile	HEX		HexRange (0 .. 0)
mod.1.sub.state	TEXT	R/O	StringRange R/O
mod.1.sub.start	TEXT	NOPRESET	StringRange
mod.1.sub.stop	TEXT	NOPRESET	StringRange
mod.1.sup.priority	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sup.symbolRate	INTEGER		IntegerRange (0 .. 0)
mod.1.sup.type	CHOICE		EnumRange (BPSK QPSK OQPSK)
mod.1.sup.filter	CHOICE	R/O	EnumRange R/O (RECTANGULAR SRRC)
mod.1.sup.sendPreamble	CHOICE	R/O	EnumRange R/O (OFF PLOP-1 PLOP-2)
mod.1.sup.preambleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sup.minIdleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sup.maxIdleLen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mod.1.sup.differential	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
mod.1.sup.convolutional	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sup.g2OutInversion	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sup.mirrorG1G2	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.1.sup.configFile	HEX		HexRange (0 .. 0)
mod.1.sup.state	TEXT	R/O	StringRange R/O
mod.1.sup.start	TEXT	NOPRESET	StringRange
mod.1.sup.stop	TEXT	NOPRESET	StringRange
dem.1.mode	CHOICE	R/O	EnumRange R/O (PM PM-DIVERSITY)
dem.1.c2noThreshold	INTEGER		IntegerRange (25 .. 60)
dem.1.frequency	FLOAT		DoubleRange (65.000000 .. 75.000000)
dem.1.acqMinLimit	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.acqMaxLimit	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.antiSideband	CHOICE	R/O	EnumRange R/O (DISABLED ENABLED)
dem.1.antiSidebandFrq1	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.antiSidebandFrq2	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.antiSidebandFrq3	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.configFile	HEX		HexRange (0 .. 0)
dem.1.state	TEXT	R/O	StringRange R/O
dem.1.meas.c2no	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
dem.1.meas.carrier	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.meas.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.meas.dopplerRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.meas.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.start	TEXT	NOPRESET	StringRange
dem.1.stop	TEXT	NOPRESET	StringRange
dem.1.unlock	TEXT	NOPRESET	StringRange
dem.1.sub.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sub.pmIndex	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sub.waveform	CHOICE	R/O	EnumRange R/O (SINE SQUARE)
dem.1.sub.symbolRate	INTEGER		IntegerRange (0 .. 0)
dem.1.sub.esn0Threshold	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sub.format	CHOICE	R/O	EnumRange R/O (NRZ SPL)
dem.1.sub.convolutional	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.g2OutInversion	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.mirrorG1G2	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.differential	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.inversion	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
dem.1.sub.turboDecoder	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.turboDerandomizer	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sub.turboRate	CHOICE	R/O	EnumRange R/O (1/2 1/3 1/4 1/6)
dem.1.sub.turboBlocks	CHOICE	R/O	EnumRange R/O (1784 3568 7136 8920)
dem.1.sub.turboLockMarkers	INTEGER	R/O	IntegerRange R/O (1 .. 5)
dem.1.sub.turboUnlockMarkers	INTEGER	R/O	IntegerRange R/O (1 .. 5)
dem.1.sub.configFile	HEX		HexRange (0 .. 0)
dem.1.sub.state	TEXT	R/O	StringRange R/O
dem.1.sub.meas.freq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sub.meas.esn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sub.meas.pmIndex	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sub.start	TEXT	NOPRESET	StringRange
dem.1.sub.stop	TEXT	NOPRESET	StringRange
dem.1.sub.unlock	TEXT	NOPRESET	StringRange
dem.1.sup.mode	CHOICE	R/O	EnumRange R/O (PM PM-DIVERSITY)
dem.1.sup.frequency	FLOAT	R/O	DoubleRange R/O (65.000000 .. 75.000000)
dem.1.sup.acqMinLimit	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)
dem.1.sup.acqMaxLimit	FLOAT	R/O	DoubleRange R/O (-400.000 .. 400.000)

name	*type*	*flags*	*range*
dem.1.sup.symbolRate	INTEGER		IntegerRange (0 .. 0)
dem.1.sup.type	CHOICE		EnumRange (BPSK QPSK OQPSK)
dem.1.sup.convolutional	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sup.g2OutInversion	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sup.mirrorG1G2	CHOICE	R/O	EnumRange R/O (OFF ON)
dem.1.sup.filter	CHOICE	R/O	EnumRange R/O (RECTANGULAR SRRC)
dem.1.sup.srrcRollOff	FLOAT	R/O	DoubleRange R/O (0.00 .. 1.00)
dem.1.sup.configFile	HEX		HexRange (0 .. 0)
dem.1.sup.state	TEXT	R/O	StringRange R/O
dem.1.sup.meas.freq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sup.meas.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sup.meas.dopplerRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sup.meas.symbolRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sup.meas.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sup.meas.esn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dem.1.sup.meas.evm	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dem.1.sup.start	TEXT	NOPRESET	StringRange
dem.1.sup.stop	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
tel.1.control	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.frameLength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.syncMarker0	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker1	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker2	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker3	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker4	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker5	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarker6	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.syncMarkerLength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxBitErrorsUnlocked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxBitErrorsLocked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxBitErrorsUnlockedChecking	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxBitErrorsLockedChecking	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxChecksUnlockedChecking	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxChecksLockedChecking	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.maxBitSlips	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.extractStrategy	CHOICE	R/O	EnumRange R/O (UNILATERAL BILATERAL)

name	*type*	*flags*	*range*
tel.1.framePolarity	CHOICE	R/O	EnumRange R/O (NORMAL COMPLELENTED BOTH)
tel.1.bitPolarityInversion	CHOICE	R/O	EnumRange R/O (STATIC-ON STATIC-OFF DYNAMIC)
tel.1.unscramble	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.unscramblePolynomial	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.unscrambleInitialization	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.reedSolomon	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.reedSolomonInterleave	INTEGER	R/O	IntegerRange R/O (1 .. 5)
tel.1.hecActive	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.crcActive	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.discardErroredFrame	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.storageActive	BOOLEAN	R/O	BooleanRange R/O (true false)
tel.1.storageFilename	HEX	R/O	HexRange R/O (0 .. 0)
tel.1.storageFormat	CHOICE	R/O	EnumRange R/O (DATA-ONLY FULL)
tel.1.storageFileEntries	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.spacecraftId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.configFile	HEX		HexRange (0 .. 0)
tel.1.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
tel.1.frameSyncState	TEXT	R/O	StringRange R/O
tel.1.masterChannelSCID	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.masterChannelOCFMismatch	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.unconfiguredMasterChannelId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.framesComplemented	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.rsErrorsCorrected	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.rsUncorrectableFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.framesForwarded	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.clcwsForwarded	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.unlock2lockTransitions	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.lock2unlockTransitions	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithoutBitErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithBitErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithoutBitSlips	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithBitSlips	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.unlock2lockTransitionsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.lock2unlockTransitionsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
tel.1.markersWithoutBitErrorsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithBitErrorsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithoutBitSlipsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.markersWithBitSlipsInv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tel.1.start	TEXT	NOPRESET	StringRange
tel.1.stop	TEXT	NOPRESET	StringRange
tel.1.update	TEXT	NOPRESET	StringRange
tel.1.updatestate	TEXT	R/O	StringRange R/O
mod.1.sendtm	HEX		HexRange (0 .. 0)
apply.bbSession	TEXT	NOPRESET	StringRange
basebandSessionNo	HEX		HexRange (0 .. 0)
cal.1.divstart	INTEGER	NOPRESET	IntegerRange (0 .. 0)
cal.1.delay	INTEGER		IntegerRange (0 .. 0)
aus.1.enable	TEXT	NOPRESET	StringRange
test.1.self	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Output power)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ext. Reference)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Remote Control)
faults.04	ALARM	R/O	AlarmFlagRange R/O (DC Calibration)

9.56 Apollo-Switch

This page describes the device driver and the device window for the Apollo Microwave

Waveguide Switching and HPA Protection system tested with 3 waveguide switches and 3+1 HPA redundancy

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
switch.01	CHOICE		EnumRange DIS (FLT A B)
switch.02	CHOICE		EnumRange DIS (FLT A B)
switch.03	CHOICE		EnumRange DIS (FLT A B)
switch.04	CHOICE		EnumRange DIS (FLT A B)
switch.05	CHOICE		EnumRange DIS (FLT A B)
switch.06	CHOICE		EnumRange DIS (FLT A B)
switch.07	CHOICE		EnumRange DIS (FLT A B)
switch.08	CHOICE		EnumRange DIS (FLT A B)
switch.09	CHOICE		EnumRange DIS (FLT A B)
switch.10	CHOICE		EnumRange DIS (FLT A B)
switch.11	CHOICE		EnumRange DIS (FLT A B)
switch.12	CHOICE		EnumRange DIS (FLT A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ext. interlock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unknown model)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HPA 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (HPA 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (HPA 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (HPA 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (HPA 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (HPA 6)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch 8)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch 9)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch 12)

9.57 Arithmetic

The Arithmetic logical device combines two numeric values by adding, subtracting, multiplying or dividing them. Both input terms may either be constant values or the actual value of any numeric parameter of another device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual input output values of the device.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Arithmetic device with the devices that deliver the input signals and to configure the device's performance.

parameter	*description*
arithmeticType	With this parameter you define the arithmetic operation the device shall perform.
comment	The comment is displayed with the device's output value at the device window. You may set it to describe the purpose of the computation.

parameter	***description***
input-A.Id	Enter the ID of the parameter the device shall use as input A here. Leave the field empty if input A shall be set to a constant value
input-A.constant	Enter the constant value for input A here. Leave the field empty if input A shall follow another parameter.
input-A.comment	The input A comment is displayed above value at the device window.
input-B.Id	Enter the ID of the parameter the device shall use as input B here. Leave the field empty if input B shall be set to a constant value
input-B.constant	Enter the constant value for input B here. Leave the field empty if input B shall follow another parameter.
input-B.comment	The input B comment is displayed above value at the device window.

Remarks

- The device uses A as the first, B as the second operand. Thus SUB calculates *a-b* , DIV calculates *a/b* .
- If a division by zero error occurs, the device does not claim this and leaves the output value unchanged.
- *Never* use the device's output parameter as one of the inputs. This feed-back will cause an endless recursion which in fact will eat up all available CPU power and memory.
- The device 'learns' the number of fraction digits to display at the main page from the connected parameter's definition. The number format of input A also is used to display the output value. If you re-configure an input from constant to parameter-ID and vice versa you may need to restart the software in order to get the number format displayed correctly.
- Connecting the device to non-numeric parameters (textual parameters, multiple choice parameters) will lead to unpredictable results.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.arithmeticType	CHOICE	SAVE SETUP	EnumRange (ADD SUB MUL DIV)
config.comment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-A.constant	TEXT	SAVE SETUP	StringRange
config.input-A.comment	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.input-B.constant	TEXT	SAVE SETUP	StringRange
config.input-B.comment	TEXT	SAVE SETUP	StringRange
output	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
a	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
b	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

9.58 Astro-EdgeStreamer

This page describes the device driver and the device window for the Astro Edge Streamer U148-X device.

It supports four RF SAT inputs and can decode 16 channels in total. Each channel can utilise one specific RF input of the existing four and have its own frequency, symbol-/code rate rolloff, modulation etc. with the ability to check the running services and stream it

Each U100 19'' rack frame supports three U148-X devices with separate MGMT IP-Address, RF inputs and Ethernet outputs. The driver treats each one is treated as a separate device wich uses its own driver.

The driver was tested with the following sw releases:

- **7100 (release 21.11.2023)**

Known issue: with the tested version, the readback time has been set to 4000ms due to the relatively long time it takes the device to update values internally. To change parameters via https, the satnms-service must have a valid login session. This is done automatically, however after every parameter set, the client logs itself out to keep the web interface useable for different operators at all time and not idle in a logged in state.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Overview of the four RF SAT inputs. Lets the user enable the different inputs and set Frequency Band / Polarisation via LNB Voltage and/or 22kHz tone. View the LNA gain and LNB voltage/current.
- --- Contains Parameters for the Channels 1 to 4 and displays power, margin, BER , programs etc.
- --- Contains Parameters for the Channels 5 to 8 and displays power, margin, BER , programs etc.
- --- Contains Parameters for the Channels 9 to 12 and displays power, margin, BER , programs etc.
- --- Contains Parameters for the Channels 13 to 16 and displays power, margin, BER , programs etc.
- --- Contains Parameters for the IP TX Streams 1 to 4 and lets the user set the channel to be transmitted as well as the IP interface and port wogether with the destination address and port.
- --- Contains Parameters for the IP TX Streams 5 to 8 and lets the user set the channel to be transmitted as well as the IP interface and port wogether with the destination address and port.
- --- Contains Parameters for the IP TX Streams 9 to 12 and lets the user set the channel to be transmitted as well as the IP interface and port wogether with the destination address and port.
- --- Contains Parameters for the IP TX Streams 13 to 16 and lets the user set the channel to be transmitted as well as the IP interface and port wogether with the destination address and port.
- --- Displays the current data rate for all Ethernet interfaces including MGMT A/B and Data A/B as well as a frame counter for each Data interface.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*

- address --- The https address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd admin-username admin-password* of the device.

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *admin-username* --- Username of the user with admin read/write rights on the device.

- *admin-password* --- Password of the user with admin read/write rights on the device needed for the M&C to emulate admin user inputs via https requests.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances. Per default it is set to 4000ms for this device.
- marginThreshold --- Defines the margin for each RX channel under which the device displays a warning or alarm.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.supply.1v2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.1v8	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.2v5	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.3v3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.5v2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.supply.13v0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.fan	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.module.r	TEXT	R/O	StringRange R/O
info.module.l	TEXT	R/O	StringRange R/O
info.cpuload.1s	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.cpuload.10s	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.1.enable	CHOICE		EnumRange (MANUAL OPEN)
input.1.status	TEXT	R/O	StringRange R/O
input.1.pol	CHOICE	NOPRESET	EnumRange (11X 11Y 12X 12Y)
input.1.vol	CHOICE	NOPRESET	EnumRange (AUTO OFF 13.0V 13.3V 13.7V 14.0V 14.3V 14.7V 15.0V 18.0V 18.3V 18.7V 19.0V 19.3V 19.7V 20.0V 20.3V)
input.1.tone	CHOICE	NOPRESET	EnumRange (AUTO ON OFF)
input.1.senvol	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.1.sencurr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.1.lnagain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.2.enable	CHOICE		EnumRange (MANUAL OPEN)
input.2.status	TEXT	R/O	StringRange R/O
input.2.pol	CHOICE	NOPRESET	EnumRange (11X 11Y 12X 12Y)

name	*type*	*flags*	*range*
input.2.vol	CHOICE	NOPRESET	EnumRange (AUTO OFF 13.0V 13.3V 13.7V 14.0V 14.3V 14.7V 15.0V 18.0V 18.3V 18.7V 19.0V 19.3V 19.7V 20.0V 20.3V)
input.2.tone	CHOICE	NOPRESET	EnumRange (AUTO ON OFF)
input.2.senvol	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.2.sencurr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.2.lnagain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.3.enable	CHOICE		EnumRange (MANUAL OPEN)
input.3.status	TEXT	R/O	StringRange R/O
input.3.pol	CHOICE	NOPRESET	EnumRange (11X 11Y 12X 12Y)
input.3.vol	CHOICE	NOPRESET	EnumRange (AUTO OFF 13.0V 13.3V 13.7V 14.0V 14.3V 14.7V 15.0V 18.0V 18.3V 18.7V 19.0V 19.3V 19.7V 20.0V 20.3V)
input.3.tone	CHOICE	NOPRESET	EnumRange (AUTO ON OFF)
input.3.senvol	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.3.sencurr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.3.lnagain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.4.enable	CHOICE		EnumRange (MANUAL OPEN)
input.4.status	TEXT	R/O	StringRange R/O
input.4.pol	CHOICE	NOPRESET	EnumRange (11X 11Y 12X 12Y)
input.4.vol	CHOICE	NOPRESET	EnumRange (AUTO OFF 13.0V 13.3V 13.7V 14.0V 14.3V 14.7V 15.0V 18.0V 18.3V 18.7V 19.0V 19.3V 19.7V 20.0V 20.3V)

name	*type*	*flags*	*range*
input.4.tone	CHOICE	NOPRESET	EnumRange (AUTO ON OFF)
input.4.senvol	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.4.sencurr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.4.lnagain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.1.name	TEXT	SAVE	StringRange
channel.1.rx.enable	CHOICE		EnumRange (OFF ON)
channel.1.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.1.freq	INTEGER		IntegerRange (0 .. 0)
channel.1.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.1.locked	TEXT	R/O	StringRange R/O
channel.1.mode	TEXT	R/O	StringRange R/O
channel.1.modulation	TEXT	R/O	StringRange R/O
channel.1.cr	TEXT	R/O	StringRange R/O
channel.1.rolloff	TEXT	R/O	StringRange R/O
channel.1.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.1.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.1.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.1.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.1.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.1.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.1.spec	TEXT	R/O	StringRange R/O
channel.1.tsid	TEXT	R/O	StringRange R/O
channel.1.onid	TEXT	R/O	StringRange R/O
channel.1.status	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
channel.2.name	TEXT	SAVE	StringRange
channel.2.rx.enable	CHOICE		EnumRange (OFF ON)
channel.2.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.2.freq	INTEGER		IntegerRange (0 .. 0)
channel.2.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.2.locked	TEXT	R/O	StringRange R/O
channel.2.mode	TEXT	R/O	StringRange R/O
channel.2.modulation	TEXT	R/O	StringRange R/O
channel.2.cr	TEXT	R/O	StringRange R/O
channel.2.rolloff	TEXT	R/O	StringRange R/O
channel.2.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.2.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.2.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.2.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.2.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.2.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.2.spec	TEXT	R/O	StringRange R/O
channel.2.tsid	TEXT	R/O	StringRange R/O
channel.2.onid	TEXT	R/O	StringRange R/O
channel.2.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.3.name	TEXT	SAVE	StringRange
channel.3.rx.enable	CHOICE		EnumRange (OFF ON)
channel.3.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.3.freq	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
channel.3.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.3.locked	TEXT	R/O	StringRange R/O
channel.3.mode	TEXT	R/O	StringRange R/O
channel.3.modulation	TEXT	R/O	StringRange R/O
channel.3.cr	TEXT	R/O	StringRange R/O
channel.3.rolloff	TEXT	R/O	StringRange R/O
channel.3.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.3.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.3.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.3.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.3.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.3.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.3.spec	TEXT	R/O	StringRange R/O
channel.3.tsid	TEXT	R/O	StringRange R/O
channel.3.onid	TEXT	R/O	StringRange R/O
channel.3.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.4.name	TEXT	SAVE	StringRange
channel.4.rx.enable	CHOICE		EnumRange (OFF ON)
channel.4.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.4.freq	INTEGER		IntegerRange (0 .. 0)
channel.4.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.4.locked	TEXT	R/O	StringRange R/O
channel.4.mode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
channel.4.modulation	TEXT	R/O	StringRange R/O
channel.4.cr	TEXT	R/O	StringRange R/O
channel.4.rolloff	TEXT	R/O	StringRange R/O
channel.4.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.4.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.4.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.4.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.4.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.4.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.4.spec	TEXT	R/O	StringRange R/O
channel.4.tsid	TEXT	R/O	StringRange R/O
channel.4.onid	TEXT	R/O	StringRange R/O
channel.4.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.5.name	TEXT	SAVE	StringRange
channel.5.rx.enable	CHOICE		EnumRange (OFF ON)
channel.5.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.5.freq	INTEGER		IntegerRange (0 .. 0)
channel.5.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.5.locked	TEXT	R/O	StringRange R/O
channel.5.mode	TEXT	R/O	StringRange R/O
channel.5.modulation	TEXT	R/O	StringRange R/O
channel.5.cr	TEXT	R/O	StringRange R/O
channel.5.rolloff	TEXT	R/O	StringRange R/O
channel.5.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
channel.5.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.5.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.5.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.5.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.5.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.5.spec	TEXT	R/O	StringRange R/O
channel.5.tsid	TEXT	R/O	StringRange R/O
channel.5.onid	TEXT	R/O	StringRange R/O
channel.5.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.6.name	TEXT	SAVE	StringRange
channel.6.rx.enable	CHOICE		EnumRange (OFF ON)
channel.6.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.6.freq	INTEGER		IntegerRange (0 .. 0)
channel.6.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.6.locked	TEXT	R/O	StringRange R/O
channel.6.mode	TEXT	R/O	StringRange R/O
channel.6.modulation	TEXT	R/O	StringRange R/O
channel.6.cr	TEXT	R/O	StringRange R/O
channel.6.rolloff	TEXT	R/O	StringRange R/O
channel.6.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.6.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.6.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.6.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
channel.6.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.6.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.6.spec	TEXT	R/O	StringRange R/O
channel.6.tsid	TEXT	R/O	StringRange R/O
channel.6.onid	TEXT	R/O	StringRange R/O
channel.6.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.7.name	TEXT	SAVE	StringRange
channel.7.rx.enable	CHOICE		EnumRange (OFF ON)
channel.7.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.7.freq	INTEGER		IntegerRange (0 .. 0)
channel.7.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.7.locked	TEXT	R/O	StringRange R/O
channel.7.mode	TEXT	R/O	StringRange R/O
channel.7.modulation	TEXT	R/O	StringRange R/O
channel.7.cr	TEXT	R/O	StringRange R/O
channel.7.rolloff	TEXT	R/O	StringRange R/O
channel.7.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.7.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.7.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.7.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.7.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.7.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.7.spec	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
channel.7.tsid	TEXT	R/O	StringRange R/O
channel.7.onid	TEXT	R/O	StringRange R/O
channel.7.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.8.name	TEXT	SAVE	StringRange
channel.8.rx.enable	CHOICE		EnumRange (OFF ON)
channel.8.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.8.freq	INTEGER		IntegerRange (0 .. 0)
channel.8.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.8.locked	TEXT	R/O	StringRange R/O
channel.8.mode	TEXT	R/O	StringRange R/O
channel.8.modulation	TEXT	R/O	StringRange R/O
channel.8.cr	TEXT	R/O	StringRange R/O
channel.8.rolloff	TEXT	R/O	StringRange R/O
channel.8.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.8.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.8.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.8.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.8.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.8.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.8.spec	TEXT	R/O	StringRange R/O
channel.8.tsid	TEXT	R/O	StringRange R/O
channel.8.onid	TEXT	R/O	StringRange R/O
channel.8.status	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
channel.9.name	TEXT	SAVE	StringRange
channel.9.rx.enable	CHOICE		EnumRange (OFF ON)
channel.9.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.9.freq	INTEGER		IntegerRange (0 .. 0)
channel.9.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.9.locked	TEXT	R/O	StringRange R/O
channel.9.mode	TEXT	R/O	StringRange R/O
channel.9.modulation	TEXT	R/O	StringRange R/O
channel.9.cr	TEXT	R/O	StringRange R/O
channel.9.rolloff	TEXT	R/O	StringRange R/O
channel.9.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.9.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.9.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.9.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.9.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.9.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.9.spec	TEXT	R/O	StringRange R/O
channel.9.tsid	TEXT	R/O	StringRange R/O
channel.9.onid	TEXT	R/O	StringRange R/O
channel.9.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.10.name	TEXT	SAVE	StringRange
channel.10.rx.enable	CHOICE		EnumRange (OFF ON)
channel.10.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.10.freq	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
channel.10.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.10.locked	TEXT	R/O	StringRange R/O
channel.10.mode	TEXT	R/O	StringRange R/O
channel.10.modulation	TEXT	R/O	StringRange R/O
channel.10.cr	TEXT	R/O	StringRange R/O
channel.10.rolloff	TEXT	R/O	StringRange R/O
channel.10.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.10.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.10.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.10.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.10.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.10.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.10.spec	TEXT	R/O	StringRange R/O
channel.10.tsid	TEXT	R/O	StringRange R/O
channel.10.onid	TEXT	R/O	StringRange R/O
channel.10.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.11.name	TEXT	SAVE	StringRange
channel.11.rx.enable	CHOICE		EnumRange (OFF ON)
channel.11.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.11.freq	INTEGER		IntegerRange (0 .. 0)
channel.11.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.11.locked	TEXT	R/O	StringRange R/O
channel.11.mode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
channel.11.modulation	TEXT	R/O	StringRange R/O
channel.11.cr	TEXT	R/O	StringRange R/O
channel.11.rolloff	TEXT	R/O	StringRange R/O
channel.11.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.11.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.11.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.11.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.11.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.11.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.11.spec	TEXT	R/O	StringRange R/O
channel.11.tsid	TEXT	R/O	StringRange R/O
channel.11.onid	TEXT	R/O	StringRange R/O
channel.11.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.12.name	TEXT	SAVE	StringRange
channel.12.rx.enable	CHOICE		EnumRange (OFF ON)
channel.12.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.12.freq	INTEGER		IntegerRange (0 .. 0)
channel.12.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.12.locked	TEXT	R/O	StringRange R/O
channel.12.mode	TEXT	R/O	StringRange R/O
channel.12.modulation	TEXT	R/O	StringRange R/O
channel.12.cr	TEXT	R/O	StringRange R/O
channel.12.rolloff	TEXT	R/O	StringRange R/O
channel.12.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
channel.12.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.12.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.12.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.12.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.12.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.12.spec	TEXT	R/O	StringRange R/O
channel.12.tsid	TEXT	R/O	StringRange R/O
channel.12.onid	TEXT	R/O	StringRange R/O
channel.12.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.13.name	TEXT	SAVE	StringRange
channel.13.rx.enable	CHOICE		EnumRange (OFF ON)
channel.13.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.13.freq	INTEGER		IntegerRange (0 .. 0)
channel.13.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.13.locked	TEXT	R/O	StringRange R/O
channel.13.mode	TEXT	R/O	StringRange R/O
channel.13.modulation	TEXT	R/O	StringRange R/O
channel.13.cr	TEXT	R/O	StringRange R/O
channel.13.rolloff	TEXT	R/O	StringRange R/O
channel.13.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.13.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.13.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.13.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
channel.13.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.13.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.13.spec	TEXT	R/O	StringRange R/O
channel.13.tsid	TEXT	R/O	StringRange R/O
channel.13.onid	TEXT	R/O	StringRange R/O
channel.13.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.14.name	TEXT	SAVE	StringRange
channel.14.rx.enable	CHOICE		EnumRange (OFF ON)
channel.14.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.14.freq	INTEGER		IntegerRange (0 .. 0)
channel.14.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.14.locked	TEXT	R/O	StringRange R/O
channel.14.mode	TEXT	R/O	StringRange R/O
channel.14.modulation	TEXT	R/O	StringRange R/O
channel.14.cr	TEXT	R/O	StringRange R/O
channel.14.rolloff	TEXT	R/O	StringRange R/O
channel.14.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.14.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.14.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.14.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.14.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.14.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.14.spec	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
channel.14.tsid	TEXT	R/O	StringRange R/O
channel.14.onid	TEXT	R/O	StringRange R/O
channel.14.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.15.name	TEXT	SAVE	StringRange
channel.15.rx.enable	CHOICE		EnumRange (OFF ON)
channel.15.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.15.freq	INTEGER		IntegerRange (0 .. 0)
channel.15.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.15.locked	TEXT	R/O	StringRange R/O
channel.15.mode	TEXT	R/O	StringRange R/O
channel.15.modulation	TEXT	R/O	StringRange R/O
channel.15.cr	TEXT	R/O	StringRange R/O
channel.15.rolloff	TEXT	R/O	StringRange R/O
channel.15.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.15.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.15.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.15.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.15.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.15.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.15.spec	TEXT	R/O	StringRange R/O
channel.15.tsid	TEXT	R/O	StringRange R/O
channel.15.onid	TEXT	R/O	StringRange R/O
channel.15.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.16.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
channel.16.rx.enable	CHOICE		EnumRange (OFF ON)
channel.16.rfinput	CHOICE		EnumRange (1 2 3 4)
channel.16.freq	INTEGER		IntegerRange (0 .. 0)
channel.16.symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
channel.16.locked	TEXT	R/O	StringRange R/O
channel.16.mode	TEXT	R/O	StringRange R/O
channel.16.modulation	TEXT	R/O	StringRange R/O
channel.16.cr	TEXT	R/O	StringRange R/O
channel.16.rolloff	TEXT	R/O	StringRange R/O
channel.16.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.16.ebn0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.16.cn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.16.cnmarg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.16.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
channel.16.berLog	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
channel.16.spec	TEXT	R/O	StringRange R/O
channel.16.tsid	TEXT	R/O	StringRange R/O
channel.16.onid	TEXT	R/O	StringRange R/O
channel.16.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.17.freq	INTEGER		IntegerRange (0 .. 0)
ipout.src.addr.a	TEXT		StringRange
ipout.src.addr.b	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.1.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.1.enable.a	CHOICE		EnumRange (ON OFF)
ipout.1.enable.b	CHOICE		EnumRange (ON OFF)
ipout.1.src.port.a	TEXT		StringRange
ipout.1.src.port.b	TEXT		StringRange
ipout.1.dest.addr.a	TEXT		StringRange
ipout.1.dest.addr.b	TEXT		StringRange
ipout.1.dest.port.a	TEXT		StringRange
ipout.1.dest.port.b	TEXT		StringRange
ipout.2.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.2.enable.a	CHOICE		EnumRange (ON OFF)
ipout.2.enable.b	CHOICE		EnumRange (ON OFF)
ipout.2.src.port.a	TEXT		StringRange
ipout.2.src.port.b	TEXT		StringRange
ipout.2.dest.addr.a	TEXT		StringRange
ipout.2.dest.addr.b	TEXT		StringRange
ipout.2.dest.port.a	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.2.dest.port.b	TEXT		StringRange
ipout.3.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.3.enable.a	CHOICE		EnumRange (ON OFF)
ipout.3.enable.b	CHOICE		EnumRange (ON OFF)
ipout.3.src.port.a	TEXT		StringRange
ipout.3.src.port.b	TEXT		StringRange
ipout.3.dest.addr.a	TEXT		StringRange
ipout.3.dest.addr.b	TEXT		StringRange
ipout.3.dest.port.a	TEXT		StringRange
ipout.3.dest.port.b	TEXT		StringRange
ipout.4.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.4.enable.a	CHOICE		EnumRange (ON OFF)
ipout.4.enable.b	CHOICE		EnumRange (ON OFF)
ipout.4.src.port.a	TEXT		StringRange
ipout.4.src.port.b	TEXT		StringRange
ipout.4.dest.addr.a	TEXT		StringRange
ipout.4.dest.addr.b	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.4.dest.port.a	TEXT		StringRange
ipout.4.dest.port.b	TEXT		StringRange
ipout.5.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.5.enable.a	CHOICE		EnumRange (ON OFF)
ipout.5.enable.b	CHOICE		EnumRange (ON OFF)
ipout.5.src.port.a	TEXT		StringRange
ipout.5.src.port.b	TEXT		StringRange
ipout.5.dest.addr.a	TEXT		StringRange
ipout.5.dest.addr.b	TEXT		StringRange
ipout.5.dest.port.a	TEXT		StringRange
ipout.5.dest.port.b	TEXT		StringRange
ipout.6.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.6.enable.a	CHOICE		EnumRange (ON OFF)
ipout.6.enable.b	CHOICE		EnumRange (ON OFF)
ipout.6.src.port.a	TEXT		StringRange
ipout.6.src.port.b	TEXT		StringRange
ipout.6.dest.addr.a	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.6.dest.addr.b	TEXT		StringRange
ipout.6.dest.port.a	TEXT		StringRange
ipout.6.dest.port.b	TEXT		StringRange
ipout.7.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.7.enable.a	CHOICE		EnumRange (ON OFF)
ipout.7.enable.b	CHOICE		EnumRange (ON OFF)
ipout.7.src.port.a	TEXT		StringRange
ipout.7.src.port.b	TEXT		StringRange
ipout.7.dest.addr.a	TEXT		StringRange
ipout.7.dest.addr.b	TEXT		StringRange
ipout.7.dest.port.a	TEXT		StringRange
ipout.7.dest.port.b	TEXT		StringRange
ipout.8.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.8.enable.a	CHOICE		EnumRange (ON OFF)
ipout.8.enable.b	CHOICE		EnumRange (ON OFF)
ipout.8.src.port.a	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.8.src.port.b	TEXT		StringRange
ipout.8.dest.addr.a	TEXT		StringRange
ipout.8.dest.addr.b	TEXT		StringRange
ipout.8.dest.port.a	TEXT		StringRange
ipout.8.dest.port.b	TEXT		StringRange
ipout.9.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.9.enable.a	CHOICE		EnumRange (ON OFF)
ipout.9.enable.b	CHOICE		EnumRange (ON OFF)
ipout.9.src.port.a	TEXT		StringRange
ipout.9.src.port.b	TEXT		StringRange
ipout.9.dest.addr.a	TEXT		StringRange
ipout.9.dest.addr.b	TEXT		StringRange
ipout.9.dest.port.a	TEXT		StringRange
ipout.9.dest.port.b	TEXT		StringRange
ipout.10.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.10.enable.a	CHOICE		EnumRange (ON OFF)
ipout.10.enable.b	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
ipout.10.src.port.a	TEXT		StringRange
ipout.10.src.port.b	TEXT		StringRange
ipout.10.dest.addr.a	TEXT		StringRange
ipout.10.dest.addr.b	TEXT		StringRange
ipout.10.dest.port.a	TEXT		StringRange
ipout.10.dest.port.b	TEXT		StringRange
ipout.11.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.11.enable.a	CHOICE		EnumRange (ON OFF)
ipout.11.enable.b	CHOICE		EnumRange (ON OFF)
ipout.11.src.port.a	TEXT		StringRange
ipout.11.src.port.b	TEXT		StringRange
ipout.11.dest.addr.a	TEXT		StringRange
ipout.11.dest.addr.b	TEXT		StringRange
ipout.11.dest.port.a	TEXT		StringRange
ipout.11.dest.port.b	TEXT		StringRange
ipout.12.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.12.enable.a	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
ipout.12.enable.b	CHOICE		EnumRange (ON OFF)
ipout.12.src.port.a	TEXT		StringRange
ipout.12.src.port.b	TEXT		StringRange
ipout.12.dest.addr.a	TEXT		StringRange
ipout.12.dest.addr.b	TEXT		StringRange
ipout.12.dest.port.a	TEXT		StringRange
ipout.12.dest.port.b	TEXT		StringRange
ipout.13.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.13.enable.a	CHOICE		EnumRange (ON OFF)
ipout.13.enable.b	CHOICE		EnumRange (ON OFF)
ipout.13.src.port.a	TEXT		StringRange
ipout.13.src.port.b	TEXT		StringRange
ipout.13.dest.addr.a	TEXT		StringRange
ipout.13.dest.addr.b	TEXT		StringRange
ipout.13.dest.port.a	TEXT		StringRange
ipout.13.dest.port.b	TEXT		StringRange
ipout.14.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)

name	*type*	*flags*	*range*
ipout.14.enable.a	CHOICE		EnumRange (ON OFF)
ipout.14.enable.b	CHOICE		EnumRange (ON OFF)
ipout.14.src.port.a	TEXT		StringRange
ipout.14.src.port.b	TEXT		StringRange
ipout.14.dest.addr.a	TEXT		StringRange
ipout.14.dest.addr.b	TEXT		StringRange
ipout.14.dest.port.a	TEXT		StringRange
ipout.14.dest.port.b	TEXT		StringRange
ipout.15.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.15.enable.a	CHOICE		EnumRange (ON OFF)
ipout.15.enable.b	CHOICE		EnumRange (ON OFF)
ipout.15.src.port.a	TEXT		StringRange
ipout.15.src.port.b	TEXT		StringRange
ipout.15.dest.addr.a	TEXT		StringRange
ipout.15.dest.addr.b	TEXT		StringRange
ipout.15.dest.port.a	TEXT		StringRange
ipout.15.dest.port.b	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.16.src	CHOICE	NOPRESET	EnumRange (UNASSIGNED CHANNEL 1 CHANNEL 2 CHANNEL 3 CHANNEL 4 CHANNEL 5 CHANNEL 6 CHANNEL 7 CHANNEL 8 CHANNEL 9 CHANNEL 10 CHANNEL 11 CHANNEL 12 CHANNEL 13 CHANNEL 14 CHANNEL 15 CHANNEL 16)
ipout.16.enable.a	CHOICE		EnumRange (ON OFF)
ipout.16.enable.b	CHOICE		EnumRange (ON OFF)
ipout.16.src.port.a	TEXT		StringRange
ipout.16.src.port.b	TEXT		StringRange
ipout.16.dest.addr.a	TEXT		StringRange
ipout.16.dest.addr.b	TEXT		StringRange
ipout.16.dest.port.a	TEXT		StringRange
ipout.16.dest.port.b	TEXT		StringRange
status.mgmt.a	TEXT		StringRange
status.mgmt.b	TEXT		StringRange
status.data.a	TEXT		StringRange
status.data.b	TEXT		StringRange
status.mgmt.a.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.mgmt.a.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.mgmt.b.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.mgmt.b.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.data.a.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
status.data.a.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.data.b.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.data.b.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
status.data.a.frames.tx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.b.frames.tx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.a.frames.rx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.b.frames.rx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.a.excep.frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.b.excep.frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.a.error.frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
status.data.b.error.frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dummy.programList	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.1.programList	CHOICE		EnumRange (NO SERVICES)
channel.2.programList	CHOICE		EnumRange (NO SERVICES)
channel.3.programList	CHOICE		EnumRange (NO SERVICES)
channel.4.programList	CHOICE		EnumRange (NO SERVICES)
channel.5.programList	CHOICE		EnumRange (NO SERVICES)
channel.6.programList	CHOICE		EnumRange (NO SERVICES)
channel.7.programList	CHOICE		EnumRange (NO SERVICES)
channel.8.programList	CHOICE		EnumRange (NO SERVICES)
channel.9.programList	CHOICE		EnumRange (NO SERVICES)
channel.10.programList	CHOICE		EnumRange (NO SERVICES)
channel.11.programList	CHOICE		EnumRange (NO SERVICES)
channel.12.programList	CHOICE		EnumRange (NO SERVICES)
channel.13.programList	CHOICE		EnumRange (NO SERVICES)

name	*type*	*flags*	*range*
channel.14.programList	CHOICE		EnumRange (NO SERVICES)
channel.15.programList	CHOICE		EnumRange (NO SERVICES)
channel.16.programList	CHOICE		EnumRange (NO SERVICES)
internal.type	TEXT	R/O	StringRange R/O
internal.power.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.power.l	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ipout.src.a.ip3	TEXT		StringRange
internal.ipout.src.a.ip2	TEXT		StringRange
internal.ipout.src.a.ip1	TEXT		StringRange
internal.ipout.src.a.ip0	TEXT		StringRange
internal.ipout.src.b.ip3	TEXT		StringRange
internal.ipout.src.b.ip2	TEXT		StringRange
internal.ipout.src.b.ip1	TEXT		StringRange
internal.ipout.src.b.ip0	TEXT		StringRange
internal.ipout.1.dest.a.ip3	TEXT		StringRange
internal.ipout.1.dest.a.ip2	TEXT		StringRange
internal.ipout.1.dest.a.ip1	TEXT		StringRange
internal.ipout.1.dest.a.ip0	TEXT		StringRange
internal.ipout.1.dest.b.ip3	TEXT		StringRange
internal.ipout.1.dest.b.ip2	TEXT		StringRange
internal.ipout.1.dest.b.ip1	TEXT		StringRange
internal.ipout.1.dest.b.ip0	TEXT		StringRange
internal.ipout.2.dest.a.ip3	TEXT		StringRange
internal.ipout.2.dest.a.ip2	TEXT		StringRange
internal.ipout.2.dest.a.ip1	TEXT		StringRange
internal.ipout.2.dest.a.ip0	TEXT		StringRange
internal.ipout.2.dest.b.ip3	TEXT		StringRange

name	*type*	*flags*	*range*
internal.ipout.2.dest.b.ip2	TEXT		StringRange
internal.ipout.2.dest.b.ip1	TEXT		StringRange
internal.ipout.2.dest.b.ip0	TEXT		StringRange
internal.ipout.3.dest.a.ip3	TEXT		StringRange
internal.ipout.3.dest.a.ip2	TEXT		StringRange
internal.ipout.3.dest.a.ip1	TEXT		StringRange
internal.ipout.3.dest.a.ip0	TEXT		StringRange
internal.ipout.3.dest.b.ip3	TEXT		StringRange
internal.ipout.3.dest.b.ip2	TEXT		StringRange
internal.ipout.3.dest.b.ip1	TEXT		StringRange
internal.ipout.3.dest.b.ip0	TEXT		StringRange
internal.ipout.4.dest.a.ip3	TEXT		StringRange
internal.ipout.4.dest.a.ip2	TEXT		StringRange
internal.ipout.4.dest.a.ip1	TEXT		StringRange
internal.ipout.4.dest.a.ip0	TEXT		StringRange
internal.ipout.4.dest.b.ip3	TEXT		StringRange
internal.ipout.4.dest.b.ip2	TEXT		StringRange
internal.ipout.4.dest.b.ip1	TEXT		StringRange
internal.ipout.4.dest.b.ip0	TEXT		StringRange
internal.ipout.5.dest.a.ip3	TEXT		StringRange
internal.ipout.5.dest.a.ip2	TEXT		StringRange
internal.ipout.5.dest.a.ip1	TEXT		StringRange
internal.ipout.5.dest.a.ip0	TEXT		StringRange
internal.ipout.5.dest.b.ip3	TEXT		StringRange
internal.ipout.5.dest.b.ip2	TEXT		StringRange
internal.ipout.5.dest.b.ip1	TEXT		StringRange

name	*type*	*flags*	*range*
internal.ipout.5.dest.b.ip0	TEXT		StringRange
internal.ipout.6.dest.a.ip3	TEXT		StringRange
internal.ipout.6.dest.a.ip2	TEXT		StringRange
internal.ipout.6.dest.a.ip1	TEXT		StringRange
internal.ipout.6.dest.a.ip0	TEXT		StringRange
internal.ipout.6.dest.b.ip3	TEXT		StringRange
internal.ipout.6.dest.b.ip2	TEXT		StringRange
internal.ipout.6.dest.b.ip1	TEXT		StringRange
internal.ipout.6.dest.b.ip0	TEXT		StringRange
internal.ipout.7.dest.a.ip3	TEXT		StringRange
internal.ipout.7.dest.a.ip2	TEXT		StringRange
internal.ipout.7.dest.a.ip1	TEXT		StringRange
internal.ipout.7.dest.a.ip0	TEXT		StringRange
internal.ipout.7.dest.b.ip3	TEXT		StringRange
internal.ipout.7.dest.b.ip2	TEXT		StringRange
internal.ipout.7.dest.b.ip1	TEXT		StringRange
internal.ipout.7.dest.b.ip0	TEXT		StringRange
internal.ipout.8.dest.a.ip3	TEXT		StringRange
internal.ipout.8.dest.a.ip2	TEXT		StringRange
internal.ipout.8.dest.a.ip1	TEXT		StringRange
internal.ipout.8.dest.a.ip0	TEXT		StringRange
internal.ipout.8.dest.b.ip3	TEXT		StringRange
internal.ipout.8.dest.b.ip2	TEXT		StringRange
internal.ipout.8.dest.b.ip1	TEXT		StringRange
internal.ipout.8.dest.b.ip0	TEXT		StringRange
internal.ipout.9.dest.a.ip3	TEXT		StringRange

name	*type*	*flags*	*range*
internal.ipout.9.dest.a.ip2	TEXT		StringRange
internal.ipout.9.dest.a.ip1	TEXT		StringRange
internal.ipout.9.dest.a.ip0	TEXT		StringRange
internal.ipout.9.dest.b.ip3	TEXT		StringRange
internal.ipout.9.dest.b.ip2	TEXT		StringRange
internal.ipout.9.dest.b.ip1	TEXT		StringRange
internal.ipout.9.dest.b.ip0	TEXT		StringRange
internal.ipout.10.dest.a.ip3	TEXT		StringRange
internal.ipout.10.dest.a.ip2	TEXT		StringRange
internal.ipout.10.dest.a.ip1	TEXT		StringRange
internal.ipout.10.dest.a.ip0	TEXT		StringRange
internal.ipout.10.dest.b.ip3	TEXT		StringRange
internal.ipout.10.dest.b.ip2	TEXT		StringRange
internal.ipout.10.dest.b.ip1	TEXT		StringRange
internal.ipout.10.dest.b.ip0	TEXT		StringRange
internal.ipout.11.dest.a.ip3	TEXT		StringRange
internal.ipout.11.dest.a.ip2	TEXT		StringRange
internal.ipout.11.dest.a.ip1	TEXT		StringRange
internal.ipout.11.dest.a.ip0	TEXT		StringRange
internal.ipout.11.dest.b.ip3	TEXT		StringRange
internal.ipout.11.dest.b.ip2	TEXT		StringRange
internal.ipout.11.dest.b.ip1	TEXT		StringRange
internal.ipout.11.dest.b.ip0	TEXT		StringRange
internal.ipout.12.dest.a.ip3	TEXT		StringRange
internal.ipout.12.dest.a.ip2	TEXT		StringRange
internal.ipout.12.dest.a.ip1	TEXT		StringRange

name	*type*	*flags*	*range*
internal.ipout.12.dest.a.ip0	TEXT		StringRange
internal.ipout.12.dest.b.ip3	TEXT		StringRange
internal.ipout.12.dest.b.ip2	TEXT		StringRange
internal.ipout.12.dest.b.ip1	TEXT		StringRange
internal.ipout.12.dest.b.ip0	TEXT		StringRange
internal.ipout.13.dest.a.ip3	TEXT		StringRange
internal.ipout.13.dest.a.ip2	TEXT		StringRange
internal.ipout.13.dest.a.ip1	TEXT		StringRange
internal.ipout.13.dest.a.ip0	TEXT		StringRange
internal.ipout.13.dest.b.ip3	TEXT		StringRange
internal.ipout.13.dest.b.ip2	TEXT		StringRange
internal.ipout.13.dest.b.ip1	TEXT		StringRange
internal.ipout.13.dest.b.ip0	TEXT		StringRange
internal.ipout.14.dest.a.ip3	TEXT		StringRange
internal.ipout.14.dest.a.ip2	TEXT		StringRange
internal.ipout.14.dest.a.ip1	TEXT		StringRange
internal.ipout.14.dest.a.ip0	TEXT		StringRange
internal.ipout.14.dest.b.ip3	TEXT		StringRange
internal.ipout.14.dest.b.ip2	TEXT		StringRange
internal.ipout.14.dest.b.ip1	TEXT		StringRange
internal.ipout.14.dest.b.ip0	TEXT		StringRange
internal.ipout.15.dest.a.ip3	TEXT		StringRange
internal.ipout.15.dest.a.ip2	TEXT		StringRange
internal.ipout.15.dest.a.ip1	TEXT		StringRange
internal.ipout.15.dest.a.ip0	TEXT		StringRange
internal.ipout.15.dest.b.ip3	TEXT		StringRange

name	*type*	*flags*	*range*
internal.ipout.15.dest.b.ip2	TEXT		StringRange
internal.ipout.15.dest.b.ip1	TEXT		StringRange
internal.ipout.15.dest.b.ip0	TEXT		StringRange
internal.ipout.16.dest.a.ip3	TEXT		StringRange
internal.ipout.16.dest.a.ip2	TEXT		StringRange
internal.ipout.16.dest.a.ip1	TEXT		StringRange
internal.ipout.16.dest.a.ip0	TEXT		StringRange
internal.ipout.16.dest.b.ip3	TEXT		StringRange
internal.ipout.16.dest.b.ip2	TEXT		StringRange
internal.ipout.16.dest.b.ip1	TEXT		StringRange
internal.ipout.16.dest.b.ip0	TEXT		StringRange
internal.channel.1.modulation	TEXT		StringRange
internal.channel.1.cr	TEXT		StringRange
internal.channel.1.helper	TEXT		StringRange
internal.channel.2.modulation	TEXT		StringRange
internal.channel.2.cr	TEXT		StringRange
internal.channel.2.helper	TEXT		StringRange
internal.channel.3.modulation	TEXT		StringRange
internal.channel.3.cr	TEXT		StringRange
internal.channel.3.helper	TEXT		StringRange
internal.channel.4.modulation	TEXT		StringRange
internal.channel.4.cr	TEXT		StringRange
internal.channel.4.helper	TEXT		StringRange
internal.channel.5.modulation	TEXT		StringRange
internal.channel.5.cr	TEXT		StringRange
internal.channel.5.helper	TEXT		StringRange

name	*type*	*flags*	*range*
internal.channel.6.modulation	TEXT		StringRange
internal.channel.6.cr	TEXT		StringRange
internal.channel.6.helper	TEXT		StringRange
internal.channel.7.modulation	TEXT		StringRange
internal.channel.7.cr	TEXT		StringRange
internal.channel.7.helper	TEXT		StringRange
internal.channel.8.modulation	TEXT		StringRange
internal.channel.8.cr	TEXT		StringRange
internal.channel.8.helper	TEXT		StringRange
internal.channel.9.modulation	TEXT		StringRange
internal.channel.9.cr	TEXT		StringRange
internal.channel.9.helper	TEXT		StringRange
internal.channel.10.modulation	TEXT		StringRange
internal.channel.10.cr	TEXT		StringRange
internal.channel.10.helper	TEXT		StringRange
internal.channel.11.modulation	TEXT		StringRange
internal.channel.11.cr	TEXT		StringRange
internal.channel.11.helper	TEXT		StringRange
internal.channel.12.modulation	TEXT		StringRange
internal.channel.12.cr	TEXT		StringRange
internal.channel.12.helper	TEXT		StringRange
internal.channel.13.modulation	TEXT		StringRange
internal.channel.13.cr	TEXT		StringRange
internal.channel.13.helper	TEXT		StringRange
internal.channel.14.modulation	TEXT		StringRange
internal.channel.14.cr	TEXT		StringRange

name	*type*	*flags*	*range*
internal.channel.14.helper	TEXT		StringRange
internal.channel.15.modulation	TEXT		StringRange
internal.channel.15.cr	TEXT		StringRange
internal.channel.15.helper	TEXT		StringRange
internal.channel.16.modulation	TEXT		StringRange
internal.channel.16.cr	TEXT		StringRange
internal.channel.16.helper	TEXT		StringRange
internal.channel.modulation	TEXT		StringRange
internal.channel.cr	TEXT		StringRange
internal.channel.helper	TEXT		StringRange
statustext	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temp 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power module R)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power module L)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Mgmt A link)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Mgmt B link)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Data A link)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Data B link)
faults.09	ALARM	R/O	AlarmFlagRange R/O (IP TX Buffer)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (RX Channel 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RX Channel 2)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RX Channel 3)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RX Channel 4)
faults.14	ALARM	R/O	AlarmFlagRange R/O (RX Channel 5)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RX Channel 6)
faults.16	ALARM	R/O	AlarmFlagRange R/O (RX Channel 7)
faults.17	ALARM	R/O	AlarmFlagRange R/O (RX Channel 8)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RX Channel 9)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RX Channel 10)
faults.20	ALARM	R/O	AlarmFlagRange R/O (RX Channel 11)
faults.21	ALARM	R/O	AlarmFlagRange R/O (RX Channel 12)
faults.22	ALARM	R/O	AlarmFlagRange R/O (RX Channel 13)
faults.23	ALARM	R/O	AlarmFlagRange R/O (RX Channel 14)
faults.24	ALARM	R/O	AlarmFlagRange R/O (RX Channel 15)
faults.25	ALARM	R/O	AlarmFlagRange R/O (RX Channel 16)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Channel 1 margin)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RX Channel 2 margin)
faults.28	ALARM	R/O	AlarmFlagRange R/O (RX Channel 3 margin)
faults.29	ALARM	R/O	AlarmFlagRange R/O (RX Channel 4 margin)
faults.30	ALARM	R/O	AlarmFlagRange R/O (RX Channel 5 margin)
faults.31	ALARM	R/O	AlarmFlagRange R/O (RX Channel 6 margin)
faults.32	ALARM	R/O	AlarmFlagRange R/O (RX Channel 7 margin)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RX Channel 8 margin)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RX Channel 9 margin)
faults.35	ALARM	R/O	AlarmFlagRange R/O (RX Channel 10 margin)
faults.36	ALARM	R/O	AlarmFlagRange R/O (RX Channel 11 margin)
faults.37	ALARM	R/O	AlarmFlagRange R/O (RX Channel 12 margin)
faults.38	ALARM	R/O	AlarmFlagRange R/O (RX Channel 13 margin)
faults.39	ALARM	R/O	AlarmFlagRange R/O (RX Channel 14 margin)
faults.40	ALARM	R/O	AlarmFlagRange R/O (RX Channel 15 margin)
faults.41	ALARM	R/O	AlarmFlagRange R/O (RX Channel 16 margin)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (High CPU load)
internal.alarmtrigger.temp	TEXT		StringRange
internal.alarmtrigger.fan	TEXT		StringRange
internal.alarmtrigger.power.r	TEXT		StringRange
internal.alarmtrigger.power.l	TEXT		StringRange
internal.alarmtrigger.mgmt.a	TEXT		StringRange
internal.alarmtrigger.mgmt.b	TEXT		StringRange
internal.alarmtrigger.data.a	TEXT		StringRange
internal.alarmtrigger.data.b	TEXT		StringRange
internal.alarmtrigger.txbuffer	TEXT		StringRange
internal.runOnce	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

9.59 Ateme-DR5000

This page describes the device driver and the device window for the Ateme DR5000 IRD, controlled via SNMP.

The Ateme IRD handles PID selection in another way than other IRDs. See paragraph "PID selection" for more information about this.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains IP in / out parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the

audio signals from the MPEG stream.

- --- Contains the parameters concerning the ASI / SDI output ports of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DR5000.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- lnbVoltage --- Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

- bandEdge --- The frequency where the software switches the LO frequencies.
- serviceHunt --- This parameter determines if the IRD will automatically choose a server / set of PIDs when tuned to a new frequency. With the Ateme IRD, this parameter also selects one of three basic operation modes for PID selection (see paragraph "PID selection" below).
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- marginOffset --- Defines an offset which will be added to the margin shown to the operator.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

PID selection

The Ateme IRD does selecting a program or a certain stream from the decoded transport stream in a different way than other IRDs do. The IRD offers three basic operation modes which do PID selection a certain manner. The configuration parameter "serviceHunt" controls this mode.

- AUTOMATIC --- In AUTOMATIC mode, the IRD automatically selects the first program in a transport stream and assigns the audio streams belonging to this program to the audio outputs one by one. The first audio stream is sent to the first embedded SDI audio and in parallel to the first AES audio output, the other audio streams are arranged accordingly. You cannot change the selected program or the audio assignments in this mode.
- AUTOSERVICE --- In AUTOSERVICE mode, the IRD tries to select the program / service which has been selected previously. No signal is output if the selected service ID is not present in the transport stream. You may select any program from the transport stream by double clicking the service name in the list on the first device window page in this mode. The audio streams for this service are selected and arranged automatically in the sam way as in AUTOMATIC mode, there is no way to change this assignment.
- PIDLOCKED --- In this mode, all decoded streams are selected manually by PID. This applies also to the video stream which only may be selected by PID in this mode, not by selecting a service. Audio streams may be selected freely by PID and arranged to the audio outputs.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.marginOffset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.hasSrt	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
internal.fpsn	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.fpsd	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.margin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.interface	CHOICE	R/O	EnumRange R/O (IP ASI SAT SRT N/A)
reset	TEXT	NOPRESET	StringRange
interface	CHOICE		EnumRange (IP ASI SAT SRT N/A)
input	CHOICE		EnumRange (1 2 3 4)
dvbmode	CHOICE		EnumRange (AUTO DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
searchRange	INTEGER		IntegerRange (0 .. 5000)
polarization	CHOICE		EnumRange (X Y)
rolloff	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
goldCode	INTEGER		IntegerRange (0 .. 262143)
multistream	CHOICE		EnumRange (OFF ON)
multistreamId	INTEGER		IntegerRange (0 .. 255)
serviceMode	CHOICE		EnumRange (AUTOMATIC AUTOSERVICE PIDLOCKED)
internal.serviceMode	CHOICE	R/O	EnumRange R/O (AUTOMATIC AUTOSERVICE PIDLOCKED)
output.asiDescramble	CHOICE		EnumRange (OFF ON)
output.conn1	CHOICE		EnumRange (AUTO-SDI SD-SDI)
output.conn2	CHOICE		EnumRange (AUTO-SDI SD-SDI)
video.pid	INTEGER		IntegerRange (0 .. 0)
video.pixelRange	CHOICE		EnumRange (FULL SMPTE)
video.outputPsf	CHOICE		EnumRange (OFF ON)
video.arDownConvert	CHOICE		EnumRange (ANAMORPHIC PILLARBOX LETTERBOX)
video.arUpConvert	CHOICE		EnumRange (ANAMORPHIC PILLARBOX LETTERBOX)
video.fail	CHOICE		EnumRange (GRAY WHITE YELLOW CYAN GREEN MAGENTA RED BLUE BLACK)
video.freezeDelay	INTEGER		IntegerRange (0 .. 0)
video.profile	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.colorType	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitDepth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.format	CHOICE	R/O	EnumRange R/O (UNKNOWN SD-NTSC-720x480i-29.97Hz SD-PAL-720x576i-25Hz HD-1280x720p-23.98Hz HD-1280x720p-24Hz HD-1280x720p-25Hz HD-1280x720p-29.97Hz HD-1280x720p-30Hz HD-1280x720p-50Hz HD-1280x720p-59.94Hz HD-1280x720p-60Hz HD-1920x1080p-23.98Hz HD-1920x1080p-24Hz HD-1920x1080p-25Hz HD-1920x1080p-29.97Hz HD-1920x1080p-30Hz HD-1920x1080p-50Hz HD-1920x1080p-59.94Hz HD-1920x1080p-60Hz HD-1920x1080i-50Hz HD-1920x1080i-59.94Hz HD-1920x1080i-60Hz HD-1920x1080psf-23.98Hz HD-1920x1080psf-24Hz HD-1920x1080psf-25Hz HD-1920x1080psf-29.97Hz HD-1920x1080psf-30Hz)
audio.1.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.1.pid	CHOICE		EnumRange (?)
audio.1.emuSilence	CHOICE		EnumRange (OFF ON)
audio.1.actMode	TEXT	R/O	StringRange R/O
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.2.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.2.pid	CHOICE		EnumRange (?)
audio.2.emuSilence	CHOICE		EnumRange (OFF ON)
audio.2.actMode	TEXT	R/O	StringRange R/O
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.3.pid	CHOICE		EnumRange (?)
audio.3.emuSilence	CHOICE		EnumRange (OFF ON)
audio.3.actMode	TEXT	R/O	StringRange R/O
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.4.pid	CHOICE		EnumRange (?)
audio.4.emuSilence	CHOICE		EnumRange (OFF ON)
audio.4.actMode	TEXT	R/O	StringRange R/O
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.5.pid	CHOICE		EnumRange (?)
audio.5.emuSilence	CHOICE		EnumRange (OFF ON)
audio.5.actMode	TEXT	R/O	StringRange R/O
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.6.pid	CHOICE		EnumRange (?)
audio.6.emuSilence	CHOICE		EnumRange (OFF ON)
audio.6.actMode	TEXT	R/O	StringRange R/O
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.7.pid	CHOICE		EnumRange (?)
audio.7.emuSilence	CHOICE		EnumRange (OFF ON)
audio.7.actMode	TEXT	R/O	StringRange R/O
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.8.pid	CHOICE		EnumRange (?)
audio.8.emuSilence	CHOICE		EnumRange (OFF ON)
audio.8.actMode	TEXT	R/O	StringRange R/O
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.9.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.9.pid	CHOICE		EnumRange (?)
audio.9.emuSilence	CHOICE		EnumRange (OFF ON)
audio.9.actMode	TEXT	R/O	StringRange R/O
audio.9.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.9.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.9.coding	TEXT	R/O	StringRange R/O
audio.9.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.10.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.10.pid	CHOICE		EnumRange (?)
audio.10.emuSilence	CHOICE		EnumRange (OFF ON)
audio.10.actMode	TEXT	R/O	StringRange R/O
audio.10.running	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
audio.10.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.10.coding	TEXT	R/O	StringRange R/O
audio.10.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.11.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.11.pid	CHOICE		EnumRange (?)
audio.11.emuSilence	CHOICE		EnumRange (OFF ON)
audio.11.actMode	TEXT	R/O	StringRange R/O
audio.11.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.11.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.11.coding	TEXT	R/O	StringRange R/O
audio.11.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.12.mode	CHOICE		EnumRange (DISABLED AUTOMATIC PASSTHROUGH PCM MONITOR DOLBYPASSTHR)
audio.12.pid	CHOICE		EnumRange (?)
audio.12.emuSilence	CHOICE		EnumRange (OFF ON)
audio.12.actMode	TEXT	R/O	StringRange R/O
audio.12.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.12.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.12.coding	TEXT	R/O	StringRange R/O
audio.12.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.scrambled	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
flags.multiProgram	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode	TEXT	R/O	StringRange R/O
state.scrambled	CHOICE	R/O	EnumRange R/O (SCRAMBLED CLEAR)
ca.bissmode	CHOICE		EnumRange (CLEAR CAM BISS-1 BISS-E1 BISS-E2 BISS2-1 BISS2-E1 BISS2-E2 BISS-CA)
ca.bisskey	TEXT		StringRange
ipin.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (0 .. 0)
ipin.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.reset	TEXT	NOPRESET	StringRange
ipin.redundancyMode	CHOICE		EnumRange (OFF AUTO TOGGLE)
ipin.2.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
ipin.2.address	TEXT		StringRange
ipin.2.port	INTEGER		IntegerRange (0 .. 0)
ipin.type	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)

name	*type*	*flags*	*range*
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.inputBer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.postFecBer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.ipPrimary	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ipSecondary	BOOLEAN	R/O	BooleanRange R/O (true false)
ipout.ipDescramble	CHOICE		EnumRange (OFF ON)
ipout.1.enable	CHOICE		EnumRange (OFF IP SRT)
ipout.1.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 0)
ipout.1.packets	INTEGER		IntegerRange (0 .. 0)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP)
ipout.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecColumns	INTEGER		IntegerRange (0 .. 0)
ipout.1.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.2.enable	CHOICE		EnumRange (OFF IP SRT)
ipout.2.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ipout.2.packets	INTEGER		IntegerRange (0 .. 0)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP)
ipout.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecColumns	INTEGER		IntegerRange (0 .. 0)
ipout.2.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.3.enable	CHOICE		EnumRange (OFF IP SRT)
ipout.3.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
ipout.3.address	TEXT		StringRange
ipout.3.port	INTEGER		IntegerRange (0 .. 0)
ipout.3.packets	INTEGER		IntegerRange (0 .. 0)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.3.protocol	CHOICE		EnumRange (UDP RTP)
ipout.3.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecColumns	INTEGER		IntegerRange (0 .. 0)
ipout.3.fecRows	INTEGER		IntegerRange (0 .. 0)
srt.redundancy	CHOICE		EnumRange (OFF ON)
srt.state	TEXT	R/O	StringRange R/O
srt.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.packetsUndecr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.timestamp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.latency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.selected	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
srt.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.1.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
srt.1.address	TEXT		StringRange
srt.1.port	INTEGER		IntegerRange (0 .. 65535)
srt.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.1.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srt.1.key	TEXT		StringRange
srt.1.tsbpd	CHOICE		EnumRange (ON OFF)
srt.1.delay	INTEGER		IntegerRange (0 .. 0)
srt.1.stripRtp	CHOICE		EnumRange (ON OFF)
srt.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.2.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
srt.2.address	TEXT		StringRange
srt.2.port	INTEGER		IntegerRange (0 .. 65535)
srt.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.2.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srt.2.key	TEXT		StringRange
srt.2.tsbpd	CHOICE		EnumRange (ON OFF)
srt.2.delay	INTEGER		IntegerRange (0 .. 0)
srt.2.stripRtp	CHOICE		EnumRange (ON OFF)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
srtout.1.address	TEXT		StringRange

name	*type*	*flags*	*range*
srtout.1.port	INTEGER		IntegerRange (0 .. 65535)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.1.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.1.key	TEXT		StringRange
srtout.1.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.1.delay	INTEGER		IntegerRange (0 .. 0)
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
srtout.2.address	TEXT		StringRange
srtout.2.port	INTEGER		IntegerRange (0 .. 65535)
srtout.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.2.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.2.key	TEXT		StringRange
srtout.2.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.2.delay	INTEGER		IntegerRange (0 .. 0)
srtout.2.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.3.interface	CHOICE		EnumRange (STREAM-1 STREAM-2)
srtout.3.address	TEXT		StringRange
srtout.3.port	INTEGER		IntegerRange (0 .. 65535)
srtout.3.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.3.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)

name	*type*	*flags*	*range*
srtout.3.key	TEXT		StringRange
srtout.3.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.3.delay	INTEGER		IntegerRange (0 .. 0)
srtout.3.overheadBw	INTEGER		IntegerRange (5 .. 100)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 5)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan 6)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 7)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 8)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.60 Ateme-DR8400

This page describes the device driver and the device window for the Ateme DR8400 IRD, controlled via SNMP.

The Ateme IRD handles PID selection in another way than other IRDs. See paragraph "PID selection" for more information about this.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains IP in / out parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- Contains the parameters concerning the ASI / SDI output ports of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DR8400.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- lnbVoltage --- Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.

- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- marginOffset --- Defines an offset which will be added to the margin shown to the operator.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

PID selection

The Ateme IRD does selecting a program or a certain stream from the decoded transport stream in a different way than other IRDs do. The IRD offers three basic operation modes which do PID selection a certain manner. The configuration parameter "serviceHunt" controls this mode.

- AUTOMATIC --- In AUTOMATIC mode, the IRD automatically selects the first program in a transport stream and assigns the audio streams belonging to this program to the audio outputs one by one. The first audio stream is sent to the first embedded SDI audio and in parallel to the first AES audio output, the other audio streams are arranged accordingly. You cannot change the selected program or the audio assignments in this mode.
- AUTOSERVICE --- In AUTOSERVICE mode, the IRD tries to select the program / service which has been selected previously. No signal is output if the selected service ID is not present in the transport stream. You may select any program from the transport stream by double clicking the service name in the list on the first device window page in this mode. The audio streams for this service are selected and arranged automatically in the sam way as in AUTOMATIC mode, there is no way to change this assignment.
- PIDLOCKED --- In this mode, all decoded streams are selected manually by PID. This applies also to the video stream which only may be selected by PID in this mode, not by selecting a service. Audio streams may be selected freely by PID and arranged to the audio outputs.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature.cpu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature.fpga	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.marginOffset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.dummy	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
internal.fpsn	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.fpsd	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.margin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.result	TEXT	R/O	StringRange R/O
internal.audioMode1	TEXT	R/O	StringRange R/O
internal.audioMode2	TEXT	R/O	StringRange R/O
internal.audioMode3	TEXT	R/O	StringRange R/O
internal.audioMode4	TEXT	R/O	StringRange R/O
internal.audioMode5	TEXT	R/O	StringRange R/O
internal.audioMode6	TEXT	R/O	StringRange R/O
internal.audioMode7	TEXT	R/O	StringRange R/O
internal.audioMode8	TEXT	R/O	StringRange R/O
internal.audioMode9	TEXT	R/O	StringRange R/O
internal.audioMode10	TEXT	R/O	StringRange R/O
internal.audioMode11	TEXT	R/O	StringRange R/O
internal.audioMode12	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.indexAudio1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.indexAudio2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.indexAudio3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.indexAudio4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.indexVideo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
interface	CHOICE		EnumRange (IP ASI SAT)
input	CHOICE		EnumRange (RF1 RF2)
dvbmode	CHOICE		EnumRange (AUTO DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (1.000 .. 70.000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
searchRange	INTEGER		IntegerRange (1000 .. 70000)
polarization	CHOICE		EnumRange (X Y)
rolloff	TEXT	R/O	StringRange R/O
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (0 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.multiProgram	BOOLEAN	R/O	BooleanRange R/O (true false)
serviceMode	CHOICE		EnumRange (AUTOMATIC AUTOSERVICE PIDLOCKED)
internal.serviceMode	CHOICE	R/O	EnumRange R/O (AUTOMATIC AUTOSERVICE PIDLOCKED)
audioList	TEXT	R/O	StringRange R/O
output.asiDescramble	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
output.conn1	CHOICE		EnumRange (AUTO HD-SDI SD-SDI ASI)
output.conn2	CHOICE		EnumRange (AUTO HD-SDI SD-SDI ASI)
output.conn3	CHOICE		EnumRange (AUTO HD-SDI SD-SDI ASI)
video.freezeDelay	INTEGER		IntegerRange (0 .. 10000)
video.pixelRange	CHOICE		EnumRange (FULL SMPTE)
video.outputPsf	CHOICE		EnumRange (OFF ON)
video.arDownConvert	CHOICE		EnumRange (ANAMORPHIC PILLARBOX LETTERBOX)
video.fail	CHOICE		EnumRange (COLOR BARS GRAY WHITE YELLOW CYAN GREEN MAGENTA RED BLUE BLACK)
video.pid	INTEGER		IntegerRange (16 .. 8190)
video.profile	CHOICE	R/O	EnumRange R/O (UNKNOWN H264 BASELINE H264 EXTENDED H264 MAIN H264 HIGH H264 HIGH 10Bit H264 HIGH 4:2:2 MP2 SIMPLE MP2 MAIN MP2 HIGH MP2 HIGH 4:2:2 MP4 SIMPLE MP4 ADV SIMPLE AAC LC AAC MAIN AAC HE AAC HE PS)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.colorType	TEXT	R/O	StringRange R/O
video.bitDepth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.coding	CHOICE		EnumRange (UNKNOWN AVC MP2v MP4v SVC VC1)

name	*type*	*flags*	*range*
audio.1.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.1.pid	INTEGER		IntegerRange (16 .. 8190)
audio.1.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.1.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.1.emuSilence	CHOICE		EnumRange (OFF ON)
audio.1.running	CHOICE	R/O	EnumRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	CHOICE	R/O	EnumRange R/O (UNKNOWN AAC AC3 AES3 EAC3 MPA)
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (16 .. 8190)
audio.2.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.2.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.2.emuSilence	CHOICE		EnumRange (OFF ON)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	CHOICE	R/O	EnumRange R/O (UNKNOWN AAC AC3 AES3 EAC3 MPA)
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.3.pid	INTEGER	R/O	IntegerRange R/O (16 .. 8190)
audio.3.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.3.pid2	INTEGER		IntegerRange (16 .. 8190)

name	*type*	*flags*	*range*
audio.3.emuSilence	CHOICE		EnumRange (OFF ON)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	CHOICE	R/O	EnumRange R/O (UNKNOWN AAC AC3 AES3 EAC3 MPA)
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.4.pid	INTEGER	R/O	IntegerRange R/O (16 .. 8190)
audio.4.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.4.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.4.emuSilence	CHOICE		EnumRange (OFF ON)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	CHOICE	R/O	EnumRange R/O (UNKNOWN AAC AC3 AES3 EAC3 MPA)
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.5.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.5.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.5.emuSilence	CHOICE		EnumRange (OFF ON)
audio.6.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.6.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.6.pid2	INTEGER		IntegerRange (16 .. 8190)

name	*type*	*flags*	*range*
audio.6.emuSilence	CHOICE		EnumRange (OFF ON)
audio.7.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.7.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.7.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.7.emuSilence	CHOICE		EnumRange (OFF ON)
audio.8.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR DOLBY-E-PASSTHROUGH)
audio.8.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.8.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.8.emuSilence	CHOICE		EnumRange (OFF ON)
audio.9.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR)
audio.9.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.9.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.9.emuSilence	CHOICE		EnumRange (OFF ON)
audio.10.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR)
audio.10.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.10.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.10.emuSilence	CHOICE		EnumRange (OFF ON)
audio.11.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR)
audio.11.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.11.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.11.emuSilence	CHOICE		EnumRange (OFF ON)
audio.12.mode	CHOICE		EnumRange (AUTOMATIC PASSTHROUGH PCM MONITOR)

name	*type*	*flags*	*range*
audio.12.pid1	INTEGER		IntegerRange (16 .. 8190)
audio.12.pid2	INTEGER		IntegerRange (16 .. 8190)
audio.12.emuSilence	CHOICE		EnumRange (OFF ON)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E1 BISS-E2)
ca.bisskey	TEXT		StringRange
ca.descrDecProgOnly	CHOICE		EnumRange (ON OFF)
asi.interface	CHOICE		EnumRange (ASI-1 ASI-2)
ipin.bufferDuration	INTEGER		IntegerRange (0 .. 999)
ipin.mode	CHOICE		EnumRange (UDP-RTP ZIXI)
ipin.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.interface	CHOICE		EnumRange (MANAGEMENT STREAM-1 STREAM-2)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 0)
ipin.2.interface	CHOICE		EnumRange (MANAGEMENT STREAM-1 STREAM-2)
ipin.2.address	TEXT		StringRange

name	*type*	*flags*	*range*
ipin.2.port	INTEGER		IntegerRange (0 .. 0)
ipin.failoverEnable	CHOICE		EnumRange (ON OFF)
ipin.failoverTriggerPeriod	INTEGER		IntegerRange (50 .. 5000)
ipin.failoverAutoJoin	CHOICE		EnumRange (ON OFF)
ipin.failoverMode	CHOICE		EnumRange (MANUAL AUTO TOGGLE)
ipin.type	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.inputBer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.postFecBer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.reset	TEXT	NOPRESET	StringRange
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.ipDescramble	CHOICE		EnumRange (OFF ON)
ipout.1.interface	CHOICE		EnumRange (MANAGEMENT STREAM-1 STREAM-2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 0)
ipout.1.packets	INTEGER		IntegerRange (1 .. 256)

name	*type*	*flags*	*range*
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP)
ipout.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.1.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.1.fecStep	INTEGER		IntegerRange (0 .. 19)
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.ipDescramble	CHOICE		EnumRange (OFF ON)
ipout.2.interface	CHOICE		EnumRange (MANAGEMENT STREAM-1 STREAM-2)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 0)
ipout.2.packets	INTEGER		IntegerRange (1 .. 256)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP)
ipout.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.2.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.2.fecStep	INTEGER		IntegerRange (0 .. 19)
ipout.3.enable	CHOICE		EnumRange (OFF ON)
ipout.3.ipDescramble	CHOICE		EnumRange (OFF ON)
ipout.3.interface	CHOICE		EnumRange (MANAGEMENT STREAM-1 STREAM-2)
ipout.3.address	TEXT		StringRange
ipout.3.port	INTEGER		IntegerRange (0 .. 0)
ipout.3.packets	INTEGER		IntegerRange (1 .. 256)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)

name	*type*	*flags*	*range*
ipout.3.protocol	CHOICE		EnumRange (UDP RTP)
ipout.3.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.3.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.3.fecStep	INTEGER		IntegerRange (0 .. 19)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.61 Ateme-Kyrion-CM5000

This page describes the device driver and the device window for the Ateme Kyrion CM5000 video encoder / modulator. The driver evaluates if a modulator module (L-Band or IF) is present at slot 6 and shows the modulator specific parameters only if a modulator module is detected.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters (only shown if a modulator was detected at slot 6.)
- --- MPEG output parameters
- --- C/A encryption parameters
- --- Video parameters

- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- Audio 5 parameters
- --- Audio 6 parameters
- --- Audio 7 parameters
- --- Audio 8 parameters
- --- Device fault list
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- loFreq --- The BUC LO frequency to be included in the displayed frequency reading. Set this to 0 if the driver shall display the transmit frequency as L-band.
- loPol --- Selects if the BUC introduces a frequency inversion. Set this to NORMAL if the LO frequency is below the transmit frequency, INVERT if the LO frequency is above.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.hasModulator	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cfgStatus	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.loFreq	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loPol	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERT)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.channel	INTEGER	SAVE SETUP	IntegerRange (1 .. 4)
config.thumb.uname	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.thumb.passwd	TEXT	SAVE SETUP	StringRange
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.type	TEXT	R/O	StringRange R/O
internal.dolby	TEXT	R/O	StringRange R/O
internal.coding	TEXT	R/O	StringRange R/O
internal.audioMode	TEXT	R/O	StringRange R/O
internal.bitDepth	CHOICE	R/O	EnumRange R/O (8-Bit 10-Bit)
internal.colorSpace	CHOICE	R/O	EnumRange R/O (4:2:0 4:2:2)
internal.feca	TEXT	R/O	StringRange R/O
internal.fecb	TEXT	R/O	StringRange R/O
internal.fecc	TEXT	R/O	StringRange R/O
internal.fecd	TEXT	R/O	StringRange R/O
internal.fece	TEXT	R/O	StringRange R/O
internal.fecf	TEXT	R/O	StringRange R/O
internal.fecg	TEXT	R/O	StringRange R/O
internal.fech	TEXT	R/O	StringRange R/O
internal.interfacea	TEXT	R/O	StringRange R/O
internal.interfaceb	TEXT	R/O	StringRange R/O
internal.interfacec	TEXT	R/O	StringRange R/O
internal.interfaced	TEXT	R/O	StringRange R/O
internal.interfacee	TEXT	R/O	StringRange R/O
internal.interfacef	TEXT	R/O	StringRange R/O
internal.interfaceg	TEXT	R/O	StringRange R/O
internal.interfaceh	TEXT	R/O	StringRange R/O
internal.cfgStatus	TEXT	R/O	StringRange R/O
internal.slotType	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
opState	CHOICE		EnumRange (STARTED STOPPED)
output.standard	CHOICE		EnumRange (MPEG DVB ATSC ISDB SBTVD-FS SBTVD-OS)
output.datarate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.nullrate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.dvbtables	CHOICE		EnumRange (OFF ON)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.netname	TEXT		StringRange
output.netid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)
output.packetsize	CHOICE		EnumRange (188 204)
output.asiEnable	CHOICE		EnumRange (OFF ON)
output.dvbEnable	CHOICE		EnumRange (OFF ON)
output.latency	CHOICE		EnumRange (HIGH-QUALITY REDUCED-LATENCY LOW-LATENCY ULTRA-LOW-LATENCY CUSTOM-ADVANCED CUSTOM-EXPERT)
mod.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
mod.rfon	CHOICE		EnumRange (OFF ON)
mod.power	FLOAT		DoubleRange (-35.0 .. 5.0)
mod.standard	CHOICE		EnumRange (DVB-S DVB-S2 DVB-DSNG)
mod.cwmode	CHOICE		EnumRange (MODULATED CW PBRS)

name	*type*	*flags*	*range*
mod.modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK 16QAM)
mod.fec	CHOICE		EnumRange (1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 7/8 8/9 9/10)
mod.symbolrate	FLOAT		DoubleRange (0.0000 .. 0.0000)
mod.spectrum	CHOICE		EnumRange (NORM INVERT)
mod.rollOff	CHOICE		EnumRange (35% 25% 20% 15% 10% 5%)
mod.pilot	CHOICE		EnumRange (OFF ON)
mod.frameSize	CHOICE		EnumRange (NORMAL SHORT)
mod.tilt	CHOICE		EnumRange (-0.040 dB/MHz -0.035 dB/MHz -0.030 dB/MHz -0.025 dB/MHz -0.020 dB/MHz -0.015 dB/MHz -0.010 dB/MHz -0.005 dB/MHz 0.000 dB/MHz +0.005 dB/MHz +0.010 dB/MHz +0.015 dB/MHz +0.020 dB/MHz +0.025 dB/MHz +0.030 dB/MHz +0.035 dB/MHz +0.040 dB/MHz)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E BISS-CA BISS-2-1 BISS-2-E)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.injectedid	TEXT	NOPRESET	StringRange
ca.injectedid2	TEXT	NOPRESET	StringRange
video.input	CHOICE		EnumRange (SDI SMPTE-2110 QUAD-3G)
video.source	CHOICE		EnumRange (525i 625i 720p50 720p59.94 720p60 1080i25 1080i29.97 1080i30 1080psf23.976 1080psf24 1080psf25 1080p23.976 1080p24 1080p25 1080p29.97 1080p30 1080p50 1080p59.94 1080p60)

name	*type*	*flags*	*range*
video.2110.source	CHOICE		EnumRange (525i 625i 720p50 720p59.94 720p60 1080i25 1080i29.97 1080i30 1080psf23.976 1080psf24 1080psf25 1080p23.976 1080p24 1080p25 1080p29.97 1080p30 1080p50 1080p59.94 1080p60 2160p23.976 2160p24 2160p25 2160p29.97 2160p30 2160p50 2160p59.94 2160p60)
video.2110.colorimetry	CHOICE		EnumRange (BT.709 BT.2020 BT.2100)
video.2110.range	CHOICE		EnumRange (NARROW NARROW-LINEAR WIDE)
video.sdiMode	CHOICE		EnumRange (AUTO FIX)
video.codec	CHOICE		EnumRange (MPEG2 H264 HEVC)
video.profile	CHOICE	NOPRESET	EnumRange (BASELINE-4:2:0-8Bit MAIN-4:2:0-8Bit HIGH-4:2:0-8Bit HIGH-4:2:0-10Bit HIGH-4:2:2-8Bit HIGH-4:2:2-10Bit HIGH-INTRA-4:2:0-10Bit HIGH-INTRA-4:2:2-10Bit MAIN-4:2:0 HIGH-4:2:0 using 4:2:2 HIGH-4:2:2)
cbc.video.profile	CHOICE		EnumRange (BASELINE-4:2:0-8Bit MAIN-4:2:0-8Bit HIGH-4:2:0-8Bit HIGH-4:2:0-10Bit HIGH-4:2:2-8Bit HIGH-4:2:2-10Bit HIGH-INTRA-4:2:0-10Bit HIGH-INTRA-4:2:2-10Bit MAIN-4:2:0 HIGH-4:2:0 using 4:2:2 HIGH-4:2:2)
video.mpeg2Level	CHOICE		EnumRange (LOW MAIN HIGH HIGH-1440)
video.rateControl	CHOICE		EnumRange (CBR CAPPED-VBR)
video.datarate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.maxrate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.quality	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
video.loss	CHOICE		EnumRange (BLACK PATTERN FREEZE CUSTOM MUTE)
video.pid	INTEGER		IntegerRange (0 .. 8190)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
ipout.1.interface	CHOICE		EnumRange (STREAM 1 STREAM 2 MGT)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.interfaceBackup	CHOICE		EnumRange (NONE STREAM 1 STREAM 2 MGT)
ipout.1.addressBackup	TEXT		StringRange
ipout.1.portBackup	INTEGER		IntegerRange (0 .. 65535)
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.protocol	CHOICE		EnumRange (RTP UDP SRT RIST)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.tos	INTEGER		IntegerRange (0 .. 2147483647)
ipout.1.packetCount	INTEGER		IntegerRange (0 .. 100)
ipout.1.stripNullPackets	CHOICE		EnumRange (OFF ON)
ipout.1.fecMode	CHOICE		EnumRange (NONE 1D 2D)
ipout.1.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.1.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.1.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.2.interface	CHOICE		EnumRange (STREAM 1 STREAM 2 MGT)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
ipout.2.interfaceBackup	CHOICE		EnumRange (NONE STREAM 1 STREAM 2 MGT)
ipout.2.addressBackup	TEXT		StringRange
ipout.2.portBackup	INTEGER		IntegerRange (0 .. 65535)
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.protocol	CHOICE		EnumRange (RTP UDP SRT RIST)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.tos	INTEGER		IntegerRange (0 .. 2147483647)
ipout.2.packetCount	INTEGER		IntegerRange (0 .. 100)
ipout.2.stripNullPackets	CHOICE		EnumRange (OFF ON)
ipout.2.fecMode	CHOICE		EnumRange (NONE 1D 2D)
ipout.2.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.2.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.2.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.3.interface	CHOICE		EnumRange (STREAM 1 STREAM 2 MGT)
ipout.3.address	TEXT		StringRange
ipout.3.port	INTEGER		IntegerRange (0 .. 65535)
ipout.3.interfaceBackup	CHOICE		EnumRange (NONE STREAM 1 STREAM 2 MGT)
ipout.3.addressBackup	TEXT		StringRange
ipout.3.portBackup	INTEGER		IntegerRange (0 .. 65535)
ipout.3.enable	CHOICE		EnumRange (OFF ON)
ipout.3.protocol	CHOICE		EnumRange (RTP UDP SRT RIST)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.3.tos	INTEGER		IntegerRange (0 .. 2147483647)
ipout.3.packetCount	INTEGER		IntegerRange (0 .. 100)

name	*type*	*flags*	*range*
ipout.3.stripNullPackets	CHOICE		EnumRange (OFF ON)
ipout.3.fecMode	CHOICE		EnumRange (NONE 1D 2D)
ipout.3.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.3.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.3.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.4.interface	CHOICE		EnumRange (STREAM 1 STREAM 2 MGT)
ipout.4.address	TEXT		StringRange
ipout.4.port	INTEGER		IntegerRange (0 .. 65535)
ipout.4.interfaceBackup	CHOICE		EnumRange (NONE STREAM 1 STREAM 2 MGT)
ipout.4.addressBackup	TEXT		StringRange
ipout.4.portBackup	INTEGER		IntegerRange (0 .. 65535)
ipout.4.enable	CHOICE		EnumRange (OFF ON)
ipout.4.protocol	CHOICE		EnumRange (RTP UDP SRT RIST)
ipout.4.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.4.tos	INTEGER		IntegerRange (0 .. 2147483647)
ipout.4.packetCount	INTEGER		IntegerRange (0 .. 100)
ipout.4.stripNullPackets	CHOICE		EnumRange (OFF ON)
ipout.4.fecMode	CHOICE		EnumRange (NONE 1D 2D)
ipout.4.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.4.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.4.fecSteps	INTEGER		IntegerRange (0 .. 20)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.1.tsbpd	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
srtout.1.latency	INTEGER		IntegerRange (0 .. 0)
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.1.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
srtout.1.key	TEXT	NOPRESET	StringRange
srtout.1.localPortBackup	INTEGER		IntegerRange (0 .. 65535)
srtout.1.streamId	TEXT		StringRange
srtout.1.bonding	CHOICE		EnumRange (ON OFF)
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.2.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.2.latency	INTEGER		IntegerRange (0 .. 0)
srtout.2.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.2.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
srtout.2.key	TEXT	NOPRESET	StringRange
srtout.2.localPortBackup	INTEGER		IntegerRange (0 .. 65535)
srtout.2.streamId	TEXT		StringRange
srtout.2.bonding	CHOICE		EnumRange (ON OFF)
srtout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.3.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.3.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.3.latency	INTEGER		IntegerRange (0 .. 0)
srtout.3.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.3.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)

name	*type*	*flags*	*range*
srtout.3.key	TEXT	NOPRESET	StringRange
srtout.3.localPortBackup	INTEGER		IntegerRange (0 .. 65535)
srtout.3.streamId	TEXT		StringRange
srtout.3.bonding	CHOICE		EnumRange (ON OFF)
srtout.4.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.4.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.4.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.4.latency	INTEGER		IntegerRange (0 .. 0)
srtout.4.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.4.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
srtout.4.key	TEXT	NOPRESET	StringRange
srtout.4.localPortBackup	INTEGER		IntegerRange (0 .. 65535)
srtout.4.streamId	TEXT		StringRange
srtout.4.bonding	CHOICE		EnumRange (ON OFF)
ristout.1.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.1.bonding	CHOICE		EnumRange (OFF BROADCAST)
ristout.2.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.2.bonding	CHOICE		EnumRange (OFF BROADCAST)
ristout.3.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.3.bonding	CHOICE		EnumRange (OFF BROADCAST)
ristout.4.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.4.bonding	CHOICE		EnumRange (OFF BROADCAST)
audio.1.enable	CHOICE		EnumRange (OFF ON)
audio.1.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.1.inputFormat	CHOICE		EnumRange (PCM AC3 AES3 DOLBY-E)

name	*type*	*flags*	*range*
audio.1.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.1.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.1.lang	TEXT		StringRange
audio.1.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.1.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.1.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.1.delay	FLOAT		DoubleRange (0 .. 0)
audio.1.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.1.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.2.enable	CHOICE		EnumRange (OFF ON)
audio.2.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.2.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.2.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.2.lang	TEXT		StringRange
audio.2.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.2.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)

name	*type*	*flags*	*range*
audio.2.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.2.delay	FLOAT		DoubleRange (0 .. 0)
audio.2.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.2.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.3.enable	CHOICE		EnumRange (OFF ON)
audio.3.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.3.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.3.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.3.lang	TEXT		StringRange
audio.3.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.3.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.3.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.3.delay	FLOAT		DoubleRange (0 .. 0)
audio.3.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.3.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)

name	*type*	*flags*	*range*
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.4.enable	CHOICE		EnumRange (OFF ON)
audio.4.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.4.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.4.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.4.lang	TEXT		StringRange
audio.4.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.4.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.4.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.4.delay	FLOAT		DoubleRange (0 .. 0)
audio.4.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.4.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.5.enable	CHOICE		EnumRange (OFF ON)
audio.5.source	CHOICE		EnumRange (ANALOG SDI AES)

name	*type*	*flags*	*range*
audio.5.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.5.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.5.lang	TEXT		StringRange
audio.5.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.5.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.5.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.5.delay	FLOAT		DoubleRange (0 .. 0)
audio.5.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.5.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.5.pid	INTEGER		IntegerRange (0 .. 8191)
audio.5.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.6.enable	CHOICE		EnumRange (OFF ON)
audio.6.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.6.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.6.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.6.lang	TEXT		StringRange
audio.6.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.6.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)

name	*type*	*flags*	*range*
audio.6.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.6.delay	FLOAT		DoubleRange (0 .. 0)
audio.6.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.6.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.6.pid	INTEGER		IntegerRange (0 .. 8191)
audio.6.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.7.enable	CHOICE		EnumRange (OFF ON)
audio.7.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.7.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.7.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.7.lang	TEXT		StringRange
audio.7.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.7.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.7.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.7.delay	FLOAT		DoubleRange (0 .. 0)
audio.7.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.7.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)

name	*type*	*flags*	*range*
audio.7.pid	INTEGER		IntegerRange (0 .. 8191)
audio.7.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
audio.8.enable	CHOICE		EnumRange (OFF ON)
audio.8.source	CHOICE		EnumRange (ANALOG SDI AES)
audio.8.samplingRate	CHOICE		EnumRange (48000 44100 32000)
audio.8.datarate	INTEGER		IntegerRange (32000 .. 640000)
audio.8.lang	TEXT		StringRange
audio.8.coding	CHOICE		EnumRange (MPEG LAYER2 PASS THROUGH)
audio.8.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.8.type	CHOICE		EnumRange (DOLBY DIGITAL DOLBY DIGITAL PLUS DOLBY E AES3/SMPTE-302M)
audio.8.delay	FLOAT		DoubleRange (0 .. 0)
audio.8.gain	FLOAT		DoubleRange (0.0 .. 0.0)
audio.8.sdiMatrix	CHOICE		EnumRange (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.8.pid	INTEGER		IntegerRange (0 .. 8191)
audio.8.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.state	CHOICE	R/O	EnumRange R/O (VALID MISSING INCOMPATIBLE INACTIVE INVALID DOLBY-E)
thumbnail	GENERIC	R/O	ObjectRange R/O generic object
faults.01	ALARM	R/O	AlarmFlagRange R/O (Modulator)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Audio 5)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Audio 6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Audio 7)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Audio 8)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Fan 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Fan 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Fan 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Fan 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.62 Ateme-TitanEdge-Encoder

This page describes the device driver and the device window for the Ateme Titan Edge Encoder, controlled via REST HTTP.

The driver is not compatible with **2.3.x** and earlier releases, It was tested with release **2.5.0 up to 2.11.5**

The driver can only handle one encoding instance. As Titan Edge server can handle more than one "encoder". Each instance has to be setup in a single device. The respective instance has to be configured at first setup via config.encoder

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general mpeg output üparameters.
- --- Contains IP output 1 settings and statistics
- --- Contains IP output 2 settings and statistics
- --- Contains IP output 3 settings and statistics
- --- Contains IP output 4 settings and statistics
- --- Contains video settings
- --- Contains audio channel 1 settings
- --- Contains audio channel 2 settings
- --- Contains audio channel 3 settings
- --- Contains audio channel 4 settings
- --- Contains audio channel 5 settings
- --- Contains audio channel 6 settings
- --- Contains audio channel 7 settings
- --- Contains audio channel 8 settings
- --- Contains the conditional access (BISS) parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Titan Edge.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added
encoder	Defines the desired gateway instance.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.channelname	TEXT	R/O	StringRange R/O
info.cpuModel	TEXT	R/O	StringRange R/O
info.cpuNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuUsage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.encoder	INTEGER	SAVE SETUP	IntegerRange (0 .. 500)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.scrambling	CHOICE	R/O	EnumRange R/O (CLEAR BISS CAM)
internal.bisskey	TEXT	R/O	StringRange R/O
internal.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.majorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.version2.7	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.version2.10	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.valid	TEXT	R/O	StringRange R/O
internal.sdiMatrix	TEXT	R/O	StringRange R/O
internal.mode.a1	TEXT	R/O	StringRange R/O
internal.mode.a2	TEXT	R/O	StringRange R/O
internal.mode.a3	TEXT	R/O	StringRange R/O
internal.mode.a4	TEXT	R/O	StringRange R/O
internal.mode.a5	TEXT	R/O	StringRange R/O
internal.mode.a6	TEXT	R/O	StringRange R/O
internal.mode.a7	TEXT	R/O	StringRange R/O
internal.mode.a8	TEXT	R/O	StringRange R/O
internal.input.a1	TEXT	R/O	StringRange R/O
internal.input.a2	TEXT	R/O	StringRange R/O
internal.input.a3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.input.a4	TEXT	R/O	StringRange R/O
internal.input.a5	TEXT	R/O	StringRange R/O
internal.input.a6	TEXT	R/O	StringRange R/O
internal.input.a7	TEXT	R/O	StringRange R/O
internal.input.a8	TEXT	R/O	StringRange R/O
internal.enhanced.a1	TEXT	R/O	StringRange R/O
internal.enhanced.a2	TEXT	R/O	StringRange R/O
internal.enhanced.a3	TEXT	R/O	StringRange R/O
internal.enhanced.a4	TEXT	R/O	StringRange R/O
internal.enhanced.a5	TEXT	R/O	StringRange R/O
internal.enhanced.a6	TEXT	R/O	StringRange R/O
internal.enhanced.a7	TEXT	R/O	StringRange R/O
internal.enhanced.a8	TEXT	R/O	StringRange R/O
internal.layout.a1	TEXT	R/O	StringRange R/O
internal.layout.a2	TEXT	R/O	StringRange R/O
internal.layout.a3	TEXT	R/O	StringRange R/O
internal.layout.a4	TEXT	R/O	StringRange R/O
internal.layout.a5	TEXT	R/O	StringRange R/O
internal.layout.a6	TEXT	R/O	StringRange R/O
internal.layout.a7	TEXT	R/O	StringRange R/O
internal.layout.a8	TEXT	R/O	StringRange R/O
internal.program.a1	TEXT	R/O	StringRange R/O
internal.program.a2	TEXT	R/O	StringRange R/O
internal.program.a3	TEXT	R/O	StringRange R/O
internal.program.a4	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.program.a5	TEXT	R/O	StringRange R/O
internal.program.a6	TEXT	R/O	StringRange R/O
internal.program.a7	TEXT	R/O	StringRange R/O
internal.program.a8	TEXT	R/O	StringRange R/O
internal.enable.a1	TEXT	R/O	StringRange R/O
internal.enable.a2	TEXT	R/O	StringRange R/O
internal.enable.a3	TEXT	R/O	StringRange R/O
internal.enable.a4	TEXT	R/O	StringRange R/O
internal.enable.a5	TEXT	R/O	StringRange R/O
internal.enable.a6	TEXT	R/O	StringRange R/O
internal.enable.a7	TEXT	R/O	StringRange R/O
internal.enable.a8	TEXT	R/O	StringRange R/O
internal.profile	TEXT	R/O	StringRange R/O
internal.color	TEXT	R/O	StringRange R/O
internal.bitdepth	TEXT	R/O	StringRange R/O
internal.datarate	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fpsNum	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fpsDen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.frameRate	TEXT	R/O	StringRange R/O
internal.cpb	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.cpbDuration	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.asiOutR	TEXT	R/O	StringRange R/O
internal.asiInR	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.sdiOutR	TEXT	R/O	StringRange R/O
internal.sdiInR	TEXT	R/O	StringRange R/O
internal.enoR	TEXT	R/O	StringRange R/O
internal.dest1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2110.format	TEXT	R/O	StringRange R/O
internal.2110.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2110.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2110.numerator	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2110.denominator	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2110.interlaced	BOOLEAN	R/O	BooleanRange R/O (true false)
output.standard	CHOICE		EnumRange (MPEG DVB ATSC ISDB SBTVD-FS SBTVD-OS)
output.datarate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.nullrate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.dvbtables	CHOICE		EnumRange (OFF ON)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.netname	TEXT		StringRange
output.serviceType	INTEGER		IntegerRange (-1 .. 255)
output.netid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)

name	*type*	*flags*	*range*
output.packetsize	CHOICE		EnumRange (188)
output.addLatency	INTEGER		IntegerRange (0 .. 1000)
output.latency	CHOICE		EnumRange (BEST-QUALITY HIGH-QUALITY REDUCED-LATENCY LOW-LATENCY)
output.autoPmtpid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.source	CHOICE		EnumRange (SDI SMPTE-2022-6 SMPTE-2110 TR-07)
video.connector	CHOICE	NOPRESET	EnumRange (?)
video.format	CHOICE		EnumRange (525i 625i 720p50 720p59.94 720p60 1080i25 1080i29.97 1080i30 1080psf23.976 1080psf24 1080psf25 1080p23.976 1080p24 1080p25 1080p29.97 1080p30 1080p50 1080p59.94 1080p60 1080plb50 1080plb59.94 1080plb60 2160p23.976 2160p24 2160p25 2160p29.97 2160p30 2160p50 2160p59.94 2160p60 2160plb50 2160plb59.94 2160plb60)
video.codec	CHOICE		EnumRange (MPEG2 H264 HEVC)
video.profile	CHOICE		EnumRange (MAIN-4:2:0-8Bit MAIN-4:2:0-10Bit MAIN-4:2:0-12Bit MAIN-4:2:2-10Bit MAIN-4:2:2-12Bit BASELINE-4:2:0-8Bit MAIN-4:2:0-8Bit HIGH-4:2:0-8Bit HIGH-4:2:0-10Bit HIGH-4:2:2-8Bit HIGH-4:2:2-10Bit HIGH-INTRA-4:2:0-10Bit HIGH-INTRA-4:2:2-10Bit MAIN-4:2:0 HIGH-4:2:2 using 4:2:0 HIGH-4:2:2)
video.rateControl	CHOICE		EnumRange (CBR CAPPED-VBR)
video.datarate	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
video.datarateMode	CHOICE		EnumRange (AUTO MANUAL)
video.maxrate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.quality	FLOAT		DoubleRange (0.0 .. 1.0)
video.loss	CHOICE		EnumRange (BLACK COLORBARS FREEZE CUSTOM MUTE)
video.pid	INTEGER		IntegerRange (0 .. 8190)
video.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.state	TEXT	R/O	StringRange R/O
video.outputFormat	CHOICE		EnumRange (AUTO CUSTOM)
video.outputResolution	CHOICE		EnumRange (1920x1080 1280x720 720x480 UNKNOWN)
video.outputFrameRate	CHOICE		EnumRange (23.976 24 25 29.97 30 50 59.94 60)
video.outputScanning	CHOICE		EnumRange (INTERLACED PROGRESSIVE)
video.tr07.format	CHOICE		EnumRange (525i 625i 720p50 720p59.94 720p60 1080i25 1080i29.97 1080i30 1080p24 1080p25 1080p29.97 1080p30 1080p50 1080p59.94 1080p60 2160p23.976 2160p24 2160p25 2160p29.97 2160p30 2160p50 2160p59.94 2160p60)
video.tr07.interface	CHOICE	NOPRESET	EnumRange (?)
video.tr07.address	TEXT		StringRange
video.tr07.port	INTEGER		IntegerRange (0 .. 65535)
video.tr07.buffer	INTEGER		IntegerRange (10 .. 999)

name	*type*	*flags*	*range*
video.2110.format	CHOICE		EnumRange (525i 625i 720p50 720p59.94 720p60 1080i25 1080i29.97 1080i30 1080p24 1080p25 1080p29.97 1080p30 1080p50 1080p59.94 1080p60 2160p23.976 2160p24 2160p25 2160p29.97 2160p30 2160p50 2160p59.94 2160p60)
video.2110.colorimetry	CHOICE		EnumRange (BT.601 BT.709 BT.2020 BT.2100 XYZ UNSPECIFIED ST240M)
video.2110.mode	CHOICE		EnumRange (2110-20 (RAW) 2110-22 (JPEG-XS))
video.2110.range	CHOICE		EnumRange (NARROW FULL)
video.2110.transferSystem	CHOICE		EnumRange (SDR PQ HLG ST428-1 UNSPECIFIED)
audio.1.enable	CHOICE		EnumRange (OFF ON)
audio.1.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.1.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.1.datarate	INTEGER		IntegerRange (32 .. 640)
audio.1.lang	TEXT		StringRange
audio.1.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.1.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)

name	*type*	*flags*	*range*
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.1.delay	FLOAT		DoubleRange (-500 .. 500)
audio.1.gain	FLOAT		DoubleRange (-15.0 .. 15.0)
audio.1.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.1.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.state	TEXT	R/O	StringRange R/O
audio.2.enable	CHOICE		EnumRange (OFF ON)
audio.2.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.2.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.2.datarate	INTEGER		IntegerRange (32 .. 640)
audio.2.lang	TEXT		StringRange
audio.2.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.2.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)

name	*type*	*flags*	*range*
audio.2.delay	FLOAT		DoubleRange (-500 .. 500)
audio.2.gain	FLOAT		DoubleRange (-15.0 .. 15.0)
audio.2.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.2.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.state	TEXT	R/O	StringRange R/O
audio.3.enable	CHOICE		EnumRange (OFF ON)
audio.3.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.3.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.3.datarate	INTEGER		IntegerRange (32 .. 640)
audio.3.lang	TEXT		StringRange
audio.3.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.3.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.3.delay	FLOAT		DoubleRange (-500 .. 500)
audio.3.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.3.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.3.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.state	TEXT	R/O	StringRange R/O
audio.4.enable	CHOICE		EnumRange (OFF ON)
audio.4.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.4.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.4.datarate	INTEGER		IntegerRange (32 .. 640)
audio.4.lang	TEXT		StringRange
audio.4.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.4.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.4.delay	FLOAT		DoubleRange (-500 .. 500)
audio.4.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.4.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.4.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.state	TEXT	R/O	StringRange R/O
audio.5.enable	CHOICE		EnumRange (OFF ON)
audio.5.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.5.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.5.datarate	INTEGER		IntegerRange (32 .. 640)
audio.5.lang	TEXT		StringRange
audio.5.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.5.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.5.pid	INTEGER		IntegerRange (0 .. 8191)
audio.5.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.5.delay	FLOAT		DoubleRange (-500 .. 500)
audio.5.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.5.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.5.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.state	TEXT	R/O	StringRange R/O
audio.6.enable	CHOICE		EnumRange (OFF ON)
audio.6.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.6.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.6.datarate	INTEGER		IntegerRange (32 .. 640)
audio.6.lang	TEXT		StringRange
audio.6.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.6.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.6.pid	INTEGER		IntegerRange (0 .. 8191)
audio.6.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.6.delay	FLOAT		DoubleRange (-500 .. 500)
audio.6.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.6.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.6.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.state	TEXT	R/O	StringRange R/O
audio.7.enable	CHOICE		EnumRange (OFF ON)
audio.7.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.7.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.7.datarate	INTEGER		IntegerRange (32 .. 640)
audio.7.lang	TEXT		StringRange
audio.7.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.7.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.7.pid	INTEGER		IntegerRange (0 .. 8191)
audio.7.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.7.delay	FLOAT		DoubleRange (-500 .. 500)
audio.7.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.7.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.7.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.state	TEXT	R/O	StringRange R/O
audio.8.enable	CHOICE		EnumRange (OFF ON)
audio.8.inputFormat	CHOICE		EnumRange (PCM (Stereo) PCM (Mono) PCM (Dual Mono) PCM (5.1) PCM (5.1.4) PCM (7.1.4) DolbyE-Program 1 DolbyE-Program 2 DolbyE-Program 3 DolbyE-Program 4 DolbyE-Program 5 DolbyE-Program 6 DolbyE-Program 7 DolbyE-Program 8)
audio.8.samplingRate	CHOICE		EnumRange (48000Hz 441000Hz 32000Hz)
audio.8.datarate	INTEGER		IntegerRange (32 .. 640)
audio.8.lang	TEXT		StringRange
audio.8.coding	CHOICE		EnumRange (MPEG LAYER2 AAC-LC-MPEG4 AAC-LC-MPEG2 AAC-HEv1 AAC-HEv2 DD DD+ AC-4 PASS THROUGH)
audio.8.mode	CHOICE		EnumRange (STEREO MONO DUAL-MONO)
audio.8.pid	INTEGER		IntegerRange (0 .. 8191)
audio.8.autoPid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.8.delay	FLOAT		DoubleRange (-500 .. 500)
audio.8.gain	FLOAT		DoubleRange (-15.0 .. 15.0)

name	*type*	*flags*	*range*
audio.8.sdiMatrix	CHOICE	R/O	EnumRange R/O (GROUP 1 - 1/2 GROUP 1 - 3/4 GROUP 2 - 1/2 GROUP 2 - 3/4 GROUP 3 - 1/2 GROUP 3 - 3/4 GROUP 4 - 1/2 GROUP 4 - 3/4)
audio.8.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.state	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
opState	CHOICE		EnumRange (STARTED STOPPED)
restart	TEXT	NOPRESET	StringRange
manualTuning	INTEGER		IntegerRange (25 .. 200)
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E BISS-CA BISS2-1 BISS2-E)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.bissCaKey	CHOICE	NOPRESET	EnumRange DIS (NONE)
ca.injectedid	CHOICE	NOPRESET	EnumRange (NONE)
ipout.1.descramble	CHOICE		EnumRange (OFF ON)
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.2.descramble	CHOICE		EnumRange (OFF ON)
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.3.descramble	CHOICE		EnumRange (OFF ON)
ipout.3.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
ipout.3.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.4.descramble	CHOICE		EnumRange (OFF ON)
ipout.4.enable	CHOICE		EnumRange (OFF ON)
ipout.4.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.1.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.1.address.1	TEXT		StringRange
ipout.1.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.1.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.1.enable.1	CHOICE		EnumRange (OFF ON)
ipout.1.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.1.address.2	TEXT		StringRange
ipout.1.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.1.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.1.enable.2	CHOICE		EnumRange (OFF ON)
ipout.2.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.2.address.1	TEXT		StringRange
ipout.2.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.2.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.2.enable.1	CHOICE		EnumRange (OFF ON)
ipout.2.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.2.address.2	TEXT		StringRange
ipout.2.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.2.tos.2	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
ipout.2.enable.2	CHOICE		EnumRange (OFF ON)
ipout.3.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.3.address.1	TEXT		StringRange
ipout.3.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.3.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.3.enable.1	CHOICE		EnumRange (OFF ON)
ipout.3.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.3.address.2	TEXT		StringRange
ipout.3.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.3.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.3.enable.2	CHOICE		EnumRange (OFF ON)
ipout.4.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.4.address.1	TEXT		StringRange
ipout.4.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.4.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.4.enable.1	CHOICE		EnumRange (OFF ON)
ipout.4.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.4.address.2	TEXT		StringRange
ipout.4.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.4.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.4.enable.2	CHOICE		EnumRange (OFF ON)
ipout.1.packets	INTEGER		IntegerRange (1 .. 100)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecColumns	INTEGER		IntegerRange (4 .. 20)

name	*type*	*flags*	*range*
ipout.1.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.1.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.2.packets	INTEGER		IntegerRange (1 .. 100)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.2.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.2.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.3.packets	INTEGER		IntegerRange (1 .. 100)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.3.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.3.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.3.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.4.packets	INTEGER		IntegerRange (1 .. 100)
ipout.4.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.4.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.4.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.4.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.4.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.4.fecSteps	INTEGER		IntegerRange (0 .. 20)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
srtout.1.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.1.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.1.key	TEXT	NOPRESET	StringRange
srtout.1.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.1.latency	INTEGER		IntegerRange (0 .. 0)
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.1.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.2.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.2.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.2.key	TEXT	NOPRESET	StringRange
srtout.2.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.2.latency	INTEGER		IntegerRange (0 .. 0)
srtout.2.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.2.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.3.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.3.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.3.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.3.key	TEXT	NOPRESET	StringRange
srtout.3.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.3.latency	INTEGER		IntegerRange (0 .. 0)
srtout.3.overheadBw	INTEGER		IntegerRange (5 .. 100)

name	*type*	*flags*	*range*
srtout.3.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.4.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.4.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.4.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.4.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.4.key	TEXT	NOPRESET	StringRange
srtout.4.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.4.latency	INTEGER		IntegerRange (0 .. 0)
srtout.4.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.4.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.1.state	TEXT	R/O	StringRange R/O
srtout.1.destination	TEXT	R/O	StringRange R/O
srtout.1.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srtout.1.list	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
srtout.1.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.uptime	TEXT	R/O	StringRange R/O
srtout.2.state	TEXT	R/O	StringRange R/O
srtout.2.destination	TEXT	R/O	StringRange R/O
srtout.2.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srtout.2.list	TEXT	R/O	StringRange R/O
srtout.2.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.uptime	TEXT	R/O	StringRange R/O
srtout.3.state	TEXT	R/O	StringRange R/O
srtout.3.destination	TEXT	R/O	StringRange R/O
srtout.3.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srtout.3.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srtout.3.list	TEXT	R/O	StringRange R/O
srtout.3.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.uptime	TEXT	R/O	StringRange R/O
srtout.4.state	TEXT	R/O	StringRange R/O
srtout.4.destination	TEXT	R/O	StringRange R/O
srtout.4.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srtout.4.list	TEXT	R/O	StringRange R/O
srtout.4.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.uptime	TEXT	R/O	StringRange R/O
ristout.1.profile	CHOICE		EnumRange (MAIN SIMPLE)

name	*type*	*flags*	*range*
ristout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.1.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.1.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.1.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.1.key	TEXT	NOPRESET	StringRange
ristout.2.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.2.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.2.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.2.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.2.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.2.key	TEXT	NOPRESET	StringRange
ristout.3.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.3.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.3.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.3.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.3.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.3.key	TEXT	NOPRESET	StringRange
ristout.4.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.4.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.4.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.4.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.4.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.4.key	TEXT	NOPRESET	StringRange
ristout.1.cname	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ristout.1.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.1.retryBw	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.cname	TEXT	R/O	StringRange R/O
ristout.2.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.2.retryBw	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.cname	TEXT	R/O	StringRange R/O
ristout.3.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ristout.3.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.3.retryBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.3.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.cname	TEXT	R/O	StringRange R/O
ristout.4.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.4.retryBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ristout.4.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
asiout.enable	CHOICE		EnumRange (OFF ON)
asiout.descramble	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
asiout.connector	CHOICE	NOPRESET	EnumRange (?)
asiout.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srtout.1.details	CHOICE	NOPRESET	EnumRange (?)
srtout.2.details	CHOICE	NOPRESET	EnumRange (?)
srtout.3.details	CHOICE	NOPRESET	EnumRange (?)
srtout.4.details	CHOICE	NOPRESET	EnumRange (?)
thumbnail	GENERIC	R/O	ObjectRange R/O generic object
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 5)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Audio 6)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Audio 7)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Audio 8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (License Missing)

9.63 Ateme-TitanEdge-Gateway

This page describes the device driver and the device window for the Ateme Titan Edge Gateway, controlled via REST HTTP.

The driver is not compatible with **2.3.x** and earlier releases, It was tested with release **2.5.0 up to 2.11.5**

The driver can only handle one gateway instance. As Titan Edge Server can handle more than one "gateway". Each instance has to be setup in a single device. The respective instance has to be configured at first setup via config.gateway

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general input and program selection.
- --- Contains the sat receiver settings and status display.
- --- Contains RTP/UDP input parameters
- --- Contains SRT input parameters
- --- Contains ZIXI input parameters
- --- Contains RIST input parameters
- --- Contains IP output 1 settings and statistics
- --- Contains IP output 2 settings and statistics
- --- Contains IP output 3 settings and statistics
- --- Contains IP output 4 settings and statistics
- --- Contains the conditional access (BISS) parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Titan Edge.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
gateway	Defines the desired gateway instance.
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.

parameter	*description*
marginThreshold	Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.channelname	TEXT	R/O	StringRange R/O
info.cpuModel	TEXT	R/O	StringRange R/O
info.cpuNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuUsage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.gateway	INTEGER	SAVE SETUP	IntegerRange (0 .. 500)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.maxDatarate	FLOAT	SETUP	DoubleRange (0.000 .. 210.000)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.lo	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.lo2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.active	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.scrambling	CHOICE	R/O	EnumRange R/O (CLEAR BISS CAM)
internal.biss	CHOICE	R/O	EnumRange R/O (BISS-1 BISS-E BISS-CA)
internal.bisskey	TEXT	R/O	StringRange R/O
internal.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.majorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.version2.7	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.version2.10	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.valid	TEXT	R/O	StringRange R/O
internal.reuse	TEXT	R/O	StringRange R/O
internal.reuse.2	TEXT	R/O	StringRange R/O
internal.asiOutR	TEXT	R/O	StringRange R/O
internal.asiInR	TEXT	R/O	StringRange R/O
internal.sdiOutR	TEXT	R/O	StringRange R/O
internal.sdiInR	TEXT	R/O	StringRange R/O
internal.inputR	TEXT	R/O	StringRange R/O
internal.input.2R	TEXT	R/O	StringRange R/O
internal.programListR	TEXT	R/O	StringRange R/O
internal.reuseR	TEXT	R/O	StringRange R/O
internal.enoR	TEXT	R/O	StringRange R/O
internal.dest1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.dest2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock.2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.scrambled	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
opState	CHOICE		EnumRange (STARTED STOPPED)
restart	TEXT	NOPRESET	StringRange
manualTuning	INTEGER		IntegerRange (50 .. 200)
interface	CHOICE		EnumRange (IP ASI SAT SRT ZIXI RIST HLS RTMP REUSE)
interface.2	CHOICE		EnumRange (IP ASI SAT SRT ZIXI RIST HLS RTMP REUSE)
input	CHOICE	NOPRESET	EnumRange (1 2 3 4)
input.2	CHOICE	NOPRESET	EnumRange (1 2 3 4)
input.active	CHOICE	R/O	EnumRange R/O (MAIN BACKUP)
input.1.active	BOOLEAN	R/O	BooleanRange R/O (true false)
input.2.active	BOOLEAN	R/O	BooleanRange R/O (true false)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
dataRate.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.redundancy	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
input.redundancyMode	CHOICE		EnumRange (AUTO TOGGLE SWITCH-ONCE MANUAL)
input.redundancyJoin	CHOICE		EnumRange (ON OFF)
input.switchover	INTEGER		IntegerRange (0 .. 0)
reuse	CHOICE		EnumRange (?)
reuse.2	CHOICE		EnumRange (?)
dvbmode	CHOICE		EnumRange (DVBS DVBS2/S2X)
cdb.dvbmode	CHOICE	NOPRESET	EnumRange (DVBS DVBS2 DVBS2X)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange	INTEGER		IntegerRange (1000 .. 50000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
rolloff	TEXT	R/O	StringRange R/O
state.ber	TEXT	R/O	StringRange R/O
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode	TEXT	R/O	StringRange R/O
dvbmode.2	CHOICE		EnumRange (DVBS DVBS2/S2X)
cdb.dvbmode.2	CHOICE	NOPRESET	EnumRange (DVBS DVBS2 DVBS2X)
frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate.2	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange.2	INTEGER		IntegerRange (1000 .. 50000)

name	*type*	*flags*	*range*
modulation.2	TEXT	R/O	StringRange R/O
fec.2	TEXT	R/O	StringRange R/O
rolloff.2	TEXT	R/O	StringRange R/O
state.ber.2	TEXT	R/O	StringRange R/O
state.snr.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ebno.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode.2	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
state.scrambled	CHOICE	R/O	EnumRange R/O (SCRAMBLED CLEAR)
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E BISS-CA CAM)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.bissCaKey	CHOICE	NOPRESET	EnumRange (NONE)
ca.injectedid	CHOICE	NOPRESET	EnumRange (NONE)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 0)
ipin.1.interface	CHOICE	NOPRESET	EnumRange (?)
ipin.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.lowfecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.vbrEnable	CHOICE		EnumRange (OFF ON)
ipin.1.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.1.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)
ipin.1.inputBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.1.postFecBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.1.jitter	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.2.address	TEXT		StringRange
ipin.2.port	INTEGER		IntegerRange (0 .. 0)
ipin.2.interface	CHOICE	NOPRESET	EnumRange (?)
ipin.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.2.lowfecEnable	CHOICE		EnumRange (OFF ON)
ipin.2.vbrEnable	CHOICE		EnumRange (OFF ON)
ipin.2.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.2.type	TEXT	R/O	StringRange R/O
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)
ipin.2.inputBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.2.postFecBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.2.jitter	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srt.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.1.interface	CHOICE	NOPRESET	EnumRange (?)
srt.1.address	TEXT		StringRange
srt.1.port	INTEGER		IntegerRange (0 .. 65535)
srt.1.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srt.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.1.latency	INTEGER		IntegerRange (0 .. 6000)
srt.1.key	TEXT	NOPRESET	StringRange
srt.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.2.interface	CHOICE	NOPRESET	EnumRange (?)
srt.2.address	TEXT		StringRange
srt.2.port	INTEGER		IntegerRange (0 .. 65535)
srt.2.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srt.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.2.latency	INTEGER		IntegerRange (0 .. 6000)
srt.2.key	TEXT	NOPRESET	StringRange
srt.1.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
srt.1.source	TEXT	R/O	StringRange R/O
srt.1.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.ackTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.nakTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsUndecr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsBelated	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srt.1.reorderDist	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.reorderTole	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.uptime	TEXT	R/O	StringRange R/O
srt.2.state	TEXT	R/O	StringRange R/O
srt.2.source	TEXT	R/O	StringRange R/O
srt.2.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.ackTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srt.2.nakTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsUndecr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsBelated	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srt.2.reorderDist	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.reorderTole	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.uptime	TEXT	R/O	StringRange R/O
zixi.1.mode	CHOICE		EnumRange (PUSH PULL)
zixi.1.interface	CHOICE	NOPRESET	EnumRange (?)
zixi.1.stream	TEXT		StringRange
zixi.1.remoteEnable	CHOICE		EnumRange (ON OFF)
zixi.1.remote	TEXT		StringRange
zixi.1.host	TEXT		StringRange
zixi.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixi.1.password	TEXT	NOPRESET	StringRange
zixi.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixi.1.decryption	CHOICE		EnumRange (DISABLED AES128 AES192 AES256)
zixi.1.key	TEXT	NOPRESET	StringRange
zixi.1.overhead	CHOICE		EnumRange (AUTO NONE 10% 20% 30% 40% 50%)
zixi.1.state	BOOLEAN	R/O	BooleanRange R/O (true false)
zixi.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
zixi.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsCorrFec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsCorrArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsDuplArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.mode	CHOICE		EnumRange (PUSH PULL)
zixi.2.interface	CHOICE	NOPRESET	EnumRange (?)
zixi.2.stream	TEXT		StringRange
zixi.2.remoteEnable	CHOICE		EnumRange (ON OFF)
zixi.2.remote	TEXT		StringRange
zixi.2.host	TEXT		StringRange
zixi.2.port	INTEGER		IntegerRange (1025 .. 65525)
zixi.2.password	TEXT	NOPRESET	StringRange
zixi.2.latency	INTEGER		IntegerRange (30 .. 10000)
zixi.2.decryption	CHOICE		EnumRange (DISABLED AES128 AES192 AES256)
zixi.2.key	TEXT	NOPRESET	StringRange
zixi.2.overhead	CHOICE		EnumRange (AUTO NONE 10% 20% 30% 40% 50%)
zixi.2.state	BOOLEAN	R/O	BooleanRange R/O (true false)
zixi.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsCorrFec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsCorrArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
zixi.2.packetsDuplArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.profile	CHOICE		EnumRange (MAIN SIMPLE)
rist.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rist.1.address	TEXT		StringRange
rist.1.port	INTEGER		IntegerRange (2 .. 65534)
rist.1.interface	CHOICE	NOPRESET	EnumRange (?)
rist.1.buffer	INTEGER		IntegerRange (100 .. 9000)
rist.1.encryption	CHOICE		EnumRange (NONE AES128 AES256)
rist.1.key	TEXT	NOPRESET	StringRange
rist.2.profile	CHOICE		EnumRange (MAIN SIMPLE)
rist.2.mode	CHOICE		EnumRange (CALLER LISTENER)
rist.2.address	TEXT		StringRange
rist.2.port	INTEGER		IntegerRange (2 .. 65534)
rist.2.interface	CHOICE	NOPRESET	EnumRange (?)
rist.2.buffer	INTEGER		IntegerRange (100 .. 9000)
rist.2.encryption	CHOICE		EnumRange (NONE AES128 AES256)
rist.2.key	TEXT	NOPRESET	StringRange
rist.1.flowId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rist.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rist.1.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.flowId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rist.2.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rist.2.lastPacketsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hls.1.address	TEXT		StringRange
hls.1.interface	CHOICE	NOPRESET	EnumRange (?)
hls.1.maxSegment	INTEGER		IntegerRange (1 .. 60)
hls.1.hasMaxBandwidth	CHOICE		EnumRange (OFF ON)
hls.1.maxBandwidth	FLOAT		DoubleRange (0.000 .. 0.000)
hls.1.video	CHOICE		EnumRange (OFF ON)
hls.1.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.videoPid	INTEGER		IntegerRange (0 .. 8191)
hls.1.useMaxBitrate	CHOICE		EnumRange (OFF ON)
hls.1.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
hls.1.audio.1	CHOICE		EnumRange (OFF ON)
hls.1.audio.2	CHOICE		EnumRange (OFF ON)
hls.1.audio.3	CHOICE		EnumRange (OFF ON)
hls.1.audio.4	CHOICE		EnumRange (OFF ON)
hls.1.audio.5	CHOICE		EnumRange (OFF ON)
hls.1.audio.6	CHOICE		EnumRange (OFF ON)
hls.1.audio.7	CHOICE		EnumRange (OFF ON)
hls.1.audio.8	CHOICE		EnumRange (OFF ON)
hls.1.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.6	INTEGER		IntegerRange (0 .. 8191)

name	*type*	*flags*	*range*
hls.1.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
hls.1.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.address	TEXT		StringRange
hls.2.interface	CHOICE	NOPRESET	EnumRange (?)
hls.2.maxSegment	INTEGER		IntegerRange (1 .. 60)
hls.2.hasMaxBandwidth	CHOICE		EnumRange (OFF ON)
hls.2.maxBandwidth	FLOAT		DoubleRange (0.000 .. 0.000)
hls.2.video	CHOICE		EnumRange (OFF ON)
hls.2.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.videoPid	INTEGER		IntegerRange (0 .. 8191)
hls.2.useMaxBitrate	CHOICE		EnumRange (OFF ON)
hls.2.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
hls.2.audio.1	CHOICE		EnumRange (OFF ON)
hls.2.audio.2	CHOICE		EnumRange (OFF ON)
hls.2.audio.3	CHOICE		EnumRange (OFF ON)
hls.2.audio.4	CHOICE		EnumRange (OFF ON)
hls.2.audio.5	CHOICE		EnumRange (OFF ON)
hls.2.audio.6	CHOICE		EnumRange (OFF ON)
hls.2.audio.7	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
hls.2.audio.8	CHOICE		EnumRange (OFF ON)
hls.2.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
hls.2.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.meas.maxSegment	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hls.2.meas.maxSegment	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rtmp.1.streamName	TEXT		StringRange
rtmp.1.interface	CHOICE	NOPRESET	EnumRange (?)
rtmp.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmp.1.source	TEXT		StringRange
rtmp.1.port	INTEGER		IntegerRange (0 .. 65535)
rtmp.1.username	TEXT		StringRange

name	*type*	*flags*	*range*
rtmp.1.password	TEXT	NOPRESET	StringRange
rtmp.1.bufferDuration	INTEGER		IntegerRange (20 .. 999)
rtmp.1.video	CHOICE		EnumRange (OFF ON)
rtmp.1.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.videoPid	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.useMaxBitrate	CHOICE		EnumRange (OFF ON)
rtmp.1.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
rtmp.1.audio.1	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.2	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.3	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.4	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.5	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.6	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.7	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.8	CHOICE		EnumRange (OFF ON)
rtmp.1.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
rtmp.1.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.streamName	TEXT		StringRange
rtmp.2.interface	CHOICE	NOPRESET	EnumRange (?)
rtmp.2.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmp.2.source	TEXT		StringRange
rtmp.2.port	INTEGER		IntegerRange (0 .. 65535)
rtmp.2.username	TEXT		StringRange
rtmp.2.password	TEXT	NOPRESET	StringRange
rtmp.2.bufferDuration	INTEGER		IntegerRange (20 .. 999)
rtmp.2.video	CHOICE		EnumRange (OFF ON)
rtmp.2.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.videoPid	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.useMaxBitrate	CHOICE		EnumRange (OFF ON)
rtmp.2.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
rtmp.2.audio.1	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.2	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.3	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.4	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.5	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.6	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.7	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.8	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
rtmp.2.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
ipout.1.descramble	CHOICE		EnumRange (OFF ON)
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.2.descramble	CHOICE		EnumRange (OFF ON)
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.3.descramble	CHOICE		EnumRange (OFF ON)
ipout.3.enable	CHOICE		EnumRange (OFF ON)
ipout.3.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)

name	*type*	*flags*	*range*
ipout.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.4.descramble	CHOICE		EnumRange (OFF ON)
ipout.4.enable	CHOICE		EnumRange (OFF ON)
ipout.4.protocol	CHOICE		EnumRange (UDP RTP SRT RIST)
ipout.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.1.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.1.address.1	TEXT		StringRange
ipout.1.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.1.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.1.enable.1	CHOICE		EnumRange (OFF ON)
ipout.1.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.1.address.2	TEXT		StringRange
ipout.1.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.1.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.1.enable.2	CHOICE		EnumRange (OFF ON)
ipout.2.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.2.address.1	TEXT		StringRange
ipout.2.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.2.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.2.enable.1	CHOICE		EnumRange (OFF ON)
ipout.2.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.2.address.2	TEXT		StringRange
ipout.2.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.2.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.2.enable.2	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
ipout.3.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.3.address.1	TEXT		StringRange
ipout.3.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.3.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.3.enable.1	CHOICE		EnumRange (OFF ON)
ipout.3.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.3.address.2	TEXT		StringRange
ipout.3.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.3.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.3.enable.2	CHOICE		EnumRange (OFF ON)
ipout.4.interface.1	CHOICE	NOPRESET	EnumRange (?)
ipout.4.address.1	TEXT		StringRange
ipout.4.port.1	INTEGER		IntegerRange (0 .. 65535)
ipout.4.tos.1	INTEGER		IntegerRange (0 .. 255)
ipout.4.enable.1	CHOICE		EnumRange (OFF ON)
ipout.4.interface.2	CHOICE	NOPRESET	EnumRange (?)
ipout.4.address.2	TEXT		StringRange
ipout.4.port.2	INTEGER		IntegerRange (0 .. 65535)
ipout.4.tos.2	INTEGER		IntegerRange (0 .. 255)
ipout.4.enable.2	CHOICE		EnumRange (OFF ON)
ipout.1.packets	INTEGER		IntegerRange (1 .. 100)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.1.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.1.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.1.fecSteps	INTEGER		IntegerRange (0 .. 20)

name	*type*	*flags*	*range*
ipout.2.packets	INTEGER		IntegerRange (1 .. 100)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.2.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.2.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.2.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.3.packets	INTEGER		IntegerRange (1 .. 100)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.3.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.3.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.3.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.3.fecSteps	INTEGER		IntegerRange (0 .. 20)
ipout.4.packets	INTEGER		IntegerRange (1 .. 100)
ipout.4.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.4.fecEnable	CHOICE		EnumRange (OFF ON)
ipout.4.fecRowEnable	CHOICE		EnumRange (OFF ON)
ipout.4.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.4.fecRows	INTEGER		IntegerRange (1 .. 20)
ipout.4.fecSteps	INTEGER		IntegerRange (0 .. 20)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.1.localPortMode	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
srtout.1.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.1.key	TEXT	NOPRESET	StringRange
srtout.1.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.1.latency	INTEGER		IntegerRange (0 .. 0)
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.1.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.2.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.2.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.2.key	TEXT	NOPRESET	StringRange
srtout.2.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.2.latency	INTEGER		IntegerRange (0 .. 0)
srtout.2.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.2.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.3.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.3.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.3.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.3.key	TEXT	NOPRESET	StringRange
srtout.3.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.3.latency	INTEGER		IntegerRange (0 .. 0)
srtout.3.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.3.maxConnect	INTEGER		IntegerRange (1 .. 1000)

name	*type*	*flags*	*range*
srtout.4.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.4.localPort	INTEGER		IntegerRange (0 .. 65535)
srtout.4.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srtout.4.encryption	CHOICE		EnumRange (NONE AES128 AES192 AES256)
srtout.4.key	TEXT	NOPRESET	StringRange
srtout.4.tsbpd	CHOICE		EnumRange (ON OFF)
srtout.4.latency	INTEGER		IntegerRange (0 .. 0)
srtout.4.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.4.maxConnect	INTEGER		IntegerRange (1 .. 1000)
srtout.1.state	TEXT	R/O	StringRange R/O
srtout.1.destination	TEXT	R/O	StringRange R/O
srtout.1.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.1.list	TEXT	R/O	StringRange R/O
srtout.1.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.uptime	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
srtout.2.state	TEXT	R/O	StringRange R/O
srtout.2.destination	TEXT	R/O	StringRange R/O
srtout.2.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.2.list	TEXT	R/O	StringRange R/O
srtout.2.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.uptime	TEXT	R/O	StringRange R/O
srtout.3.state	TEXT	R/O	StringRange R/O
srtout.3.destination	TEXT	R/O	StringRange R/O
srtout.3.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srtout.3.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.3.list	TEXT	R/O	StringRange R/O
srtout.3.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.uptime	TEXT	R/O	StringRange R/O
srtout.4.state	TEXT	R/O	StringRange R/O
srtout.4.destination	TEXT	R/O	StringRange R/O
srtout.4.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.ackRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.nakRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.4.list	TEXT	R/O	StringRange R/O
srtout.4.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.uptime	TEXT	R/O	StringRange R/O
ristout.1.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.1.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.1.buffer	INTEGER		IntegerRange (100 .. 9000)

name	*type*	*flags*	*range*
ristout.1.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.1.key	TEXT	NOPRESET	StringRange
ristout.2.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.2.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.2.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.2.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.2.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.2.key	TEXT	NOPRESET	StringRange
ristout.3.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.3.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.3.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.3.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.3.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.3.key	TEXT	NOPRESET	StringRange
ristout.4.profile	CHOICE		EnumRange (MAIN SIMPLE)
ristout.4.mode	CHOICE		EnumRange (CALLER LISTENER)
ristout.4.payloadSize	INTEGER		IntegerRange (188 .. 9986)
ristout.4.buffer	INTEGER		IntegerRange (100 .. 9000)
ristout.4.encryption	CHOICE		EnumRange (NONE AES128 AES256)
ristout.4.key	TEXT	NOPRESET	StringRange
ristout.1.cname	TEXT	R/O	StringRange R/O
ristout.1.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.1.retryBw	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ristout.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.1.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.cname	TEXT	R/O	StringRange R/O
ristout.2.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.2.retryBw	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.2.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.cname	TEXT	R/O	StringRange R/O
ristout.3.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.3.retryBw	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.3.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ristout.3.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.3.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.cname	TEXT	R/O	StringRange R/O
ristout.4.peerId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.4.retryBw	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ristout.4.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ristout.4.lastPacketsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
asiout.enable	CHOICE		EnumRange (OFF ON)
asiout.descramble	CHOICE		EnumRange (OFF ON)
asiout.connector	CHOICE	NOPRESET	EnumRange (?)
asiout.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.1.details	CHOICE	NOPRESET	EnumRange (?)
srtout.2.details	CHOICE	NOPRESET	EnumRange (?)
srtout.3.details	CHOICE	NOPRESET	EnumRange (?)
srtout.4.details	CHOICE	NOPRESET	EnumRange (?)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (RF Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.05	ALARM	R/O	AlarmFlagRange R/O (License)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.64 Ateme-TitanEdge-IRD

This page describes the device driver and the device window for the Ateme Titan Edge IRD, controlled via REST HTTP.

The driver is not compatible with **2.3.x** and earlier releases, It was tested with release **2.5.0 up to 2.11.5**

The driver can only handle one decoding instance. As Titan Edge Server can handle more than one "Decoder". Each instance has to be setup in a single device. The respective instance has to be configured at first setup via config.decoder

The Ateme IRD handles PID selection in another way than other IRDs. See paragraph "PID selection" for more information about this.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general input and program selection.
- --- Contains the sat receiver settings and status display.
- --- Contains RTP/UDP input parameters
- --- Contains SRT input parameters
- --- Contains ZIXI input parameters
- --- Contains RIST input parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Titan Edge.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
decoder	Defines the desired decoding instance.
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.
marginThreshold	Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

PID selection

The Ateme Titan Edge does selecting a program or a certain stream from the decoded transport stream in a different way than other IRDs do. The IRD offers three basic operation modes which do PID selection a certain manner. The configuration parameter "serviceHunt" controls this mode.

- AUTOMATIC --- In AUTOMATIC mode, the IRD automatically selects the first program in a transport stream and assigns the audio streams belonging to this program to the audio outputs one by one. The first audio stream is sent to the first embedded SDI, the other audio streams are arranged accordingly. You cannot change the selected program or the audio assignments in this mode.
- AUTOSERVICE --- In AUTOSERVICE mode, the IRD tries to select the program / service

which has been selected previously. No signal is output if the selected service ID is not present in the transport stream. You may select any program from the transport stream by double clicking the service name in the list on the first device window page in this mode. The audio streams for this service are selected and arranged automatically in the same way as in AUTOMATIC mode, there is no way to change this assignment.
- PIDLOCKED --- In this mode, all decoded streams are selected manually by PID. This applies also to the video stream which only may be selected by PID in this mode, not by selecting a service. Audio streams may be selected freely by PID and arranged to the audio outputs.
- AUTOMAPPING --- In AUTOMAPPING mode, the IRD automatically selects the first program in a transport stream and assigns the audio streams belonging to this program to the audio outputs one by one. When AUTOMAPPING composition mode is selected, all audio modes for all audio outputs will be set by the decoder

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.channelname	TEXT	R/O	StringRange R/O
info.cpuModel	TEXT	R/O	StringRange R/O
info.cpuNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuUsage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.decoder	INTEGER	SAVE SETUP	IntegerRange (0 .. 500)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.maxDatarate	FLOAT	SETUP	DoubleRange (0.000 .. 210.000)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 20000)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.lo	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.lo2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.fpsNum	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fpsDen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.active	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.scrambling	CHOICE	R/O	EnumRange R/O (CLEAR BISS CAM)
internal.biss	CHOICE	R/O	EnumRange R/O (BISS-1 BISS-E BISS-CA)
internal.bisskey	TEXT	R/O	StringRange R/O
internal.serviceMode	CHOICE	R/O	EnumRange R/O (AUTOMATIC AUTOSERVICE AUTOMAPPING PIDLOCKED)
internal.video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
internal.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.programNoRw	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.programNoRo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.programAuto	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.majorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minorVersion	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.version2.7	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.version2.10	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.valid	TEXT	R/O	StringRange R/O
internal.reuse	TEXT	R/O	StringRange R/O
internal.reuse.2	TEXT	R/O	StringRange R/O
internal.programListR	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.reuseR	TEXT	R/O	StringRange R/O
internal.asiOutR	TEXT	R/O	StringRange R/O
internal.asiInR	TEXT	R/O	StringRange R/O
internal.sdiOutR	TEXT	R/O	StringRange R/O
internal.sdiInR	TEXT	R/O	StringRange R/O
internal.inputR	TEXT	R/O	StringRange R/O
internal.input.2R	TEXT	R/O	StringRange R/O
internal.enoR	TEXT	R/O	StringRange R/O
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock.2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.scrambled	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
opState	CHOICE		EnumRange (STARTED STOPPED)
restart	TEXT	NOPRESET	StringRange
manualTuning	INTEGER		IntegerRange (50 .. 200)
interface	CHOICE		EnumRange (IP ASI SAT SRT ZIXI RIST HLS RTMP REUSE)
interface.2	CHOICE		EnumRange (IP ASI SAT SRT ZIXI RIST HLS RTMP REUSE)
input	CHOICE	NOPRESET	EnumRange (1 2 3 4)
input.2	CHOICE	NOPRESET	EnumRange (1 2 3 4)
input.active	CHOICE	R/O	EnumRange R/O (MAIN BACKUP)

name	*type*	*flags*	*range*
input.1.active	BOOLEAN	R/O	BooleanRange R/O (true false)
input.2.active	BOOLEAN	R/O	BooleanRange R/O (true false)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
dataRate.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.redundancy	CHOICE		EnumRange (ON OFF)
input.redundancyMode	CHOICE		EnumRange (AUTO TOGGLE SWITCH-ONCE MANUAL)
input.redundancyJoin	CHOICE		EnumRange (ON OFF)
input.switchover	INTEGER		IntegerRange (0 .. 0)
reuse	CHOICE		EnumRange (?)
reuse.2	CHOICE		EnumRange (?)
dvbmode	CHOICE		EnumRange (DVBS DVBS2/S2X)
cdb.dvbmode	CHOICE	NOPRESET	EnumRange (DVBS DVBS2 DVBS2X)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange	INTEGER		IntegerRange (1000 .. 50000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
rolloff	TEXT	R/O	StringRange R/O
state.ber	TEXT	R/O	StringRange R/O
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode	TEXT	R/O	StringRange R/O
dvbmode.2	CHOICE		EnumRange (DVBS DVBS2/S2X)
cdb.dvbmode.2	CHOICE	NOPRESET	EnumRange (DVBS DVBS2 DVBS2X)

name	*type*	*flags*	*range*
frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate.2	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange.2	INTEGER		IntegerRange (1000 .. 50000)
modulation.2	TEXT	R/O	StringRange R/O
fec.2	TEXT	R/O	StringRange R/O
rolloff.2	TEXT	R/O	StringRange R/O
state.ber.2	TEXT	R/O	StringRange R/O
state.snr.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode.2	TEXT	R/O	StringRange R/O
programNo	INTEGER		IntegerRange (0 .. 65535)
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
serviceMode	CHOICE		EnumRange (AUTOMATIC AUTOSERVICE AUTOMAPPING PIDLOCKED)
tsOrder	CHOICE		EnumRange (FOLLOW PID FOLLOW PMT)
state.scrambled	CHOICE	R/O	EnumRange R/O (SCRAMBLED CLEAR)
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E BISS-CA CAM)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.bissCaKey	CHOICE	NOPRESET	EnumRange (NONE Factory)
ca.injectedid	CHOICE	NOPRESET	EnumRange (NONE)
ipin.1.address	TEXT		StringRange

name	*type*	*flags*	*range*
ipin.1.port	INTEGER		IntegerRange (0 .. 0)
ipin.1.interface	CHOICE	NOPRESET	EnumRange (?)
ipin.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.lowfecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.vbrEnable	CHOICE		EnumRange (OFF ON)
ipin.1.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.1.type	TEXT	R/O	StringRange R/O
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)
ipin.1.inputBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.1.postFecBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.1.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.address	TEXT		StringRange
ipin.2.port	INTEGER		IntegerRange (0 .. 0)
ipin.2.interface	CHOICE	NOPRESET	EnumRange (?)
ipin.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.2.lowfecEnable	CHOICE		EnumRange (OFF ON)
ipin.2.vbrEnable	CHOICE		EnumRange (OFF ON)
ipin.2.bufferDuration	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ipin.2.type	TEXT	R/O	StringRange R/O
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)
ipin.2.inputBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.2.postFecBer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ipin.2.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.1.interface	CHOICE	NOPRESET	EnumRange (?)
srt.1.address	TEXT		StringRange
srt.1.port	INTEGER		IntegerRange (0 .. 65535)
srt.1.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srt.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.1.latency	INTEGER		IntegerRange (0 .. 6000)
srt.1.key	TEXT	NOPRESET	StringRange
srt.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srt.2.interface	CHOICE	NOPRESET	EnumRange (?)

name	*type*	*flags*	*range*
srt.2.address	TEXT		StringRange
srt.2.port	INTEGER		IntegerRange (0 .. 65535)
srt.2.localPortMode	CHOICE		EnumRange (AUTO MANUAL)
srt.2.localPort	INTEGER		IntegerRange (0 .. 65535)
srt.2.latency	INTEGER		IntegerRange (0 .. 6000)
srt.2.key	TEXT	NOPRESET	StringRange
srt.1.state	TEXT	R/O	StringRange R/O
srt.1.source	TEXT	R/O	StringRange R/O
srt.1.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.ackTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.nakTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsUndecr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.packetsBelated	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srt.1.reorderDist	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.reorderTole	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.1.uptime	TEXT	R/O	StringRange R/O
srt.2.state	TEXT	R/O	StringRange R/O
srt.2.source	TEXT	R/O	StringRange R/O
srt.2.meas.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srt.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.ackTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.nakTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsUndecr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.packetsBelated	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
srt.2.reorderDist	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.reorderTole	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srt.2.uptime	TEXT	R/O	StringRange R/O
zixi.1.mode	CHOICE		EnumRange (PUSH PULL)
zixi.1.interface	CHOICE	NOPRESET	EnumRange (?)
zixi.1.stream	TEXT		StringRange
zixi.1.remoteEnable	CHOICE		EnumRange (ON OFF)
zixi.1.remote	TEXT		StringRange
zixi.1.host	TEXT		StringRange
zixi.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixi.1.password	TEXT	NOPRESET	StringRange
zixi.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixi.1.decryption	CHOICE		EnumRange (DISABLED AES128 AES192 AES256)
zixi.1.key	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
zixi.1.overhead	CHOICE		EnumRange (AUTO NONE 10% 20% 30% 40% 50%)
zixi.1.state	BOOLEAN	R/O	BooleanRange R/O (true false)
zixi.1.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsCorrFec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsCorrArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.1.packetsDuplArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.mode	CHOICE		EnumRange (PUSH PULL)
zixi.2.interface	CHOICE	NOPRESET	EnumRange (?)
zixi.2.stream	TEXT		StringRange
zixi.2.remoteEnable	CHOICE		EnumRange (ON OFF)
zixi.2.remote	TEXT		StringRange
zixi.2.host	TEXT		StringRange
zixi.2.port	INTEGER		IntegerRange (1025 .. 65525)
zixi.2.password	TEXT	NOPRESET	StringRange
zixi.2.latency	INTEGER		IntegerRange (30 .. 10000)
zixi.2.decryption	CHOICE		EnumRange (DISABLED AES128 AES192 AES256)
zixi.2.key	TEXT	NOPRESET	StringRange
zixi.2.overhead	CHOICE		EnumRange (AUTO NONE 10% 20% 30% 40% 50%)
zixi.2.state	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
zixi.2.reconnect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsCorrFec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsCorrArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
zixi.2.packetsDuplArq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.profile	CHOICE		EnumRange (MAIN SIMPLE)
rist.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rist.1.address	TEXT		StringRange
rist.1.port	INTEGER		IntegerRange (2 .. 65534)
rist.1.interface	CHOICE	NOPRESET	EnumRange (?)
rist.1.buffer	INTEGER		IntegerRange (100 .. 9000)
rist.1.encryption	CHOICE		EnumRange (NONE AES128 AES256)
rist.1.key	TEXT	NOPRESET	StringRange
rist.2.profile	CHOICE		EnumRange (MAIN SIMPLE)
rist.2.mode	CHOICE		EnumRange (CALLER LISTENER)
rist.2.address	TEXT		StringRange
rist.2.port	INTEGER		IntegerRange (2 .. 65534)
rist.2.interface	CHOICE	NOPRESET	EnumRange (?)
rist.2.buffer	INTEGER		IntegerRange (100 .. 9000)
rist.2.encryption	CHOICE		EnumRange (NONE AES128 AES256)
rist.2.key	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
rist.1.flowId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rist.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.1.lastPacketsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.flowId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rist.2.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rist.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsMiss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsRecoRe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rist.2.lastPacketsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hls.1.address	TEXT		StringRange
hls.1.interface	CHOICE	NOPRESET	EnumRange (?)
hls.1.maxSegment	INTEGER		IntegerRange (1 .. 60)
hls.1.hasMaxBandwidth	CHOICE		EnumRange (OFF ON)
hls.1.maxBandwidth	FLOAT		DoubleRange (0.000 .. 0.000)
hls.1.video	CHOICE		EnumRange (OFF ON)
hls.1.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.videoPid	INTEGER		IntegerRange (0 .. 8191)
hls.1.useMaxBitrate	CHOICE		EnumRange (OFF ON)
hls.1.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
hls.1.audio.1	CHOICE		EnumRange (OFF ON)
hls.1.audio.2	CHOICE		EnumRange (OFF ON)
hls.1.audio.3	CHOICE		EnumRange (OFF ON)
hls.1.audio.4	CHOICE		EnumRange (OFF ON)
hls.1.audio.5	CHOICE		EnumRange (OFF ON)
hls.1.audio.6	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
hls.1.audio.7	CHOICE		EnumRange (OFF ON)
hls.1.audio.8	CHOICE		EnumRange (OFF ON)
hls.1.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
hls.1.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
hls.1.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.1.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.address	TEXT		StringRange
hls.2.interface	CHOICE	NOPRESET	EnumRange (?)
hls.2.maxSegment	INTEGER		IntegerRange (1 .. 60)
hls.2.hasMaxBandwidth	CHOICE		EnumRange (OFF ON)
hls.2.maxBandwidth	FLOAT		DoubleRange (0.000 .. 0.000)
hls.2.video	CHOICE		EnumRange (OFF ON)
hls.2.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
hls.2.videoPid	INTEGER		IntegerRange (0 .. 8191)
hls.2.useMaxBitrate	CHOICE		EnumRange (OFF ON)
hls.2.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
hls.2.audio.1	CHOICE		EnumRange (OFF ON)
hls.2.audio.2	CHOICE		EnumRange (OFF ON)
hls.2.audio.3	CHOICE		EnumRange (OFF ON)
hls.2.audio.4	CHOICE		EnumRange (OFF ON)
hls.2.audio.5	CHOICE		EnumRange (OFF ON)
hls.2.audio.6	CHOICE		EnumRange (OFF ON)
hls.2.audio.7	CHOICE		EnumRange (OFF ON)
hls.2.audio.8	CHOICE		EnumRange (OFF ON)
hls.2.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
hls.2.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
hls.2.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
hls.2.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
hls.1.meas.maxSegment	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hls.2.meas.maxSegment	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rtmp.1.streamName	TEXT		StringRange
rtmp.1.interface	CHOICE	NOPRESET	EnumRange (?)
rtmp.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmp.1.source	TEXT		StringRange
rtmp.1.port	INTEGER		IntegerRange (0 .. 65535)
rtmp.1.username	TEXT		StringRange
rtmp.1.password	TEXT		StringRange
rtmp.1.bufferDuration	INTEGER		IntegerRange (20 .. 999)
rtmp.1.video	CHOICE		EnumRange (OFF ON)
rtmp.1.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.videoPid	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.useMaxBitrate	CHOICE		EnumRange (OFF ON)
rtmp.1.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
rtmp.1.audio.1	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.2	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.3	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.4	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.5	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.6	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.7	CHOICE		EnumRange (OFF ON)
rtmp.1.audio.8	CHOICE		EnumRange (OFF ON)
rtmp.1.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.3	INTEGER		IntegerRange (0 .. 8191)

name	*type*	*flags*	*range*
rtmp.1.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
rtmp.1.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.1.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.streamName	TEXT		StringRange
rtmp.2.interface	CHOICE	NOPRESET	EnumRange (?)
rtmp.2.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmp.2.source	TEXT		StringRange
rtmp.2.port	INTEGER		IntegerRange (0 .. 65535)
rtmp.2.username	TEXT		StringRange
rtmp.2.password	TEXT		StringRange
rtmp.2.bufferDuration	INTEGER		IntegerRange (20 .. 999)
rtmp.2.video	CHOICE		EnumRange (OFF ON)
rtmp.2.useVideoPid	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.videoPid	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.useMaxBitrate	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
rtmp.2.maxBitrate	FLOAT		DoubleRange (0.100 .. 210.000)
rtmp.2.audio.1	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.2	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.3	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.4	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.5	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.6	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.7	CHOICE		EnumRange (OFF ON)
rtmp.2.audio.8	CHOICE		EnumRange (OFF ON)
rtmp.2.audioPid.1	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.2	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.3	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.4	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.5	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.6	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.7	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.audioPid.8	INTEGER		IntegerRange (0 .. 8191)
rtmp.2.useAudioPid.1	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.2	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.3	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.4	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.5	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.6	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.7	CHOICE	R/O	EnumRange R/O (OFF ON)
rtmp.2.useAudioPid.8	CHOICE	R/O	EnumRange R/O (OFF ON)
video.pid	INTEGER		IntegerRange (0 .. 8191)

name	*type*	*flags*	*range*
video.fail	CHOICE		EnumRange (BLACK COLOR BARS CUSTOM)
video.freezeDelay	INTEGER		IntegerRange (0 .. 0)
video.profile	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.colorType	TEXT	R/O	StringRange R/O
video.aspectRatio	TEXT	R/O	StringRange R/O
video.numerator	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.denominator	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitDepth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.1.pid	CHOICE		EnumRange (?)
audio.1.emuSilence	CHOICE		EnumRange (OFF ON)
audio.1.delay	INTEGER		IntegerRange (-500 .. 500)
audio.1.actMode	TEXT	R/O	StringRange R/O
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.2.pid	CHOICE		EnumRange (?)

name	*type*	*flags*	*range*
audio.2.emuSilence	CHOICE		EnumRange (OFF ON)
audio.2.delay	INTEGER		IntegerRange (-500 .. 500)
audio.2.actMode	TEXT	R/O	StringRange R/O
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.3.pid	CHOICE		EnumRange (?)
audio.3.emuSilence	CHOICE		EnumRange (OFF ON)
audio.3.delay	INTEGER		IntegerRange (-500 .. 500)
audio.3.actMode	TEXT	R/O	StringRange R/O
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.4.pid	CHOICE		EnumRange (?)
audio.4.emuSilence	CHOICE		EnumRange (OFF ON)
audio.4.delay	INTEGER		IntegerRange (-500 .. 500)
audio.4.actMode	TEXT	R/O	StringRange R/O
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.5.pid	CHOICE		EnumRange (?)
audio.5.emuSilence	CHOICE		EnumRange (OFF ON)
audio.5.delay	INTEGER		IntegerRange (-500 .. 500)
audio.5.actMode	TEXT	R/O	StringRange R/O
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.6.pid	CHOICE		EnumRange (?)
audio.6.emuSilence	CHOICE		EnumRange (OFF ON)
audio.6.delay	INTEGER		IntegerRange (-500 .. 500)
audio.6.actMode	TEXT	R/O	StringRange R/O
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.7.pid	CHOICE		EnumRange (?)
audio.7.emuSilence	CHOICE		EnumRange (OFF ON)
audio.7.delay	INTEGER		IntegerRange (-500 .. 500)

name	*type*	*flags*	*range*
audio.7.actMode	TEXT	R/O	StringRange R/O
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.mode	CHOICE		EnumRange (DISABLED AUTO PCM MONITOR PASSTHROUGH DOLBYE-PASS)
audio.8.pid	CHOICE		EnumRange (?)
audio.8.emuSilence	CHOICE		EnumRange (OFF ON)
audio.8.delay	INTEGER		IntegerRange (-500 .. 500)
audio.8.actMode	TEXT	R/O	StringRange R/O
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
thumbnail	GENERIC	R/O	ObjectRange R/O generic object
faults.01	ALARM	R/O	AlarmFlagRange R/O (RF Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.05	ALARM	R/O	AlarmFlagRange R/O (License)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.65 Aten-VGA-Splitter

This page describes the device driver and the device window for a Aten VGA Splitter with up to 8-outputs (Models VS0102, VS0104 and VS0108). The driver doesn´t support switching of

cascaded VGA Splitters. It allows to switch on/off, mute and reset each output independently.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Output Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Aten VGA Splitter do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
output.01.video	CHOICE		EnumRange (ON OFF)
output.01.audio	CHOICE		EnumRange (ON OFF)
output.01.monitor	CHOICE	R/O	EnumRange R/O (true false)
output.01.reset	TEXT	NOPRESET	StringRange
output.02.video	CHOICE		EnumRange (ON OFF)
output.02.audio	CHOICE		EnumRange (ON OFF)
output.02.monitor	CHOICE	R/O	EnumRange R/O (true false)
output.02.reset	TEXT	NOPRESET	StringRange
output.03.video	CHOICE		EnumRange DIS (OFF ON)
output.03.audio	CHOICE		EnumRange DIS (OFF ON)
output.03.monitor	CHOICE	R/O	EnumRange R/O DIS (false true)
output.03.reset	TEXT	NOPRESET	StringRange DIS
output.04.video	CHOICE		EnumRange DIS (ON OFF)
output.04.audio	CHOICE		EnumRange DIS (ON OFF)
output.04.monitor	CHOICE	R/O	EnumRange R/O DIS (true false)
output.04.reset	TEXT	NOPRESET	StringRange DIS
output.05.video	CHOICE		EnumRange DIS (ON OFF)
output.05.audio	CHOICE		EnumRange DIS (ON OFF)
output.05.monitor	CHOICE	R/O	EnumRange R/O DIS (true false)
output.05.reset	TEXT	NOPRESET	StringRange DIS
output.06.video	CHOICE		EnumRange DIS (ON OFF)
output.06.audio	CHOICE		EnumRange DIS (ON OFF)
output.06.monitor	CHOICE	R/O	EnumRange R/O DIS (true false)
output.06.reset	TEXT	NOPRESET	StringRange DIS

name	*type*	*flags*	*range*
output.07.video	CHOICE		EnumRange DIS (ON OFF)
output.07.audio	CHOICE		EnumRange DIS (ON OFF)
output.07.monitor	CHOICE	R/O	EnumRange R/O DIS (true false)
output.07.reset	TEXT	NOPRESET	StringRange DIS
output.08.video	CHOICE		EnumRange DIS (ON OFF)
output.08.audio	CHOICE		EnumRange DIS (ON OFF)
output.08.monitor	CHOICE	R/O	EnumRange R/O DIS (true false)
output.08.reset	TEXT	NOPRESET	StringRange DIS
edid	CHOICE		EnumRange (AUTO PORT 1 DEFAULT)
reset	TEXT	NOPRESET	StringRange
outputs.video	CHOICE		EnumRange (ON OFF)
outputs.audio	CHOICE		EnumRange (ON OFF)

9.66 AtlanTecRF-TLT

This page describes the *AtlanTecRF-TLT* device driver and its device window via AtlanTec-ASCII protocol. It was developed for AtlanTecRF (Atlantic Microwave/ETLSystems) Test Loop Translator of ALR series with synthesized LO.

It was tested with the Ka-band model ALR-00000000-Ka

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- RX --- Transmit parameters.
- --- Meter readings. (Shows no data for this TLT model)
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	empty
accessMode	Select Local or Remote access

The TLT may be controlled through it´s Ethernet interface. The default tcp port number for this is 4000. Assuming the TLT is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:4000" in the sat-nms VLC setup in order to make the software talk to the TLT.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.freqHigh	INTEGER		IntegerRange (0 .. 0)
info.freqLow	INTEGER		IntegerRange (0 .. 0)
info.attnHigh	INTEGER		IntegerRange (0 .. 0)
info.attnLow	INTEGER		IntegerRange (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.accessMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.attn	FLOAT		DoubleRange (0.00 .. 30.00)
lo.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
internal.attnHigh	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.attnLow	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.attnStep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.attn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Frequency lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TX ON)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX OFF)

9.67 AtlanticSat-ASC300

This page describes the device driver and the device window for the ASC Atlantic Satellite Corporation beacon receivers of series 3xx

You have to select the model name to adapts the frequency range. The driver was tested via Ethernet, but should also work on serial RS232 port.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
model	The beacon receiver model
Serial	Remember to set the line parameters for the serial interface (only if connected to serial port of beacon receiver and not to Ethernet)

Ethernet interface

The Ethernet port of ASC3xx is realized with an internal Moxa N-Port server. Therefore you have to configure this N-Port server with the Moxa software tool to "TCP Server Mode" before you can connect the sat-nms MNC system.

1. Reset the Ethernet on the back of the unit and start all over (take about 10sec)
2. Unit is set to remote and address is set to 3 for now
3. Make sure both switches on the rear need to be set to Enet and RS232
4. Open NPort search utility and click Search and the assigned OP address should appear
5. Highlight IP and click on Assign IP and enter the desired IP address (do not use DHCP)
6. Open the webinterface of beacon/receiver = NPort server
7. Go to Operating Settings / Port 1 and set Operation Mode to "TCP Server Mode" and keep the default settings except Delimiter 1 and 2 are both set to 0. Keep the port number 4001.
8. Go to Serial Settings / Port 1 and set Baudrate=2400, data bits=8, stop bits=1, parity=none and Flow control=none. None.(it was RTS/CTS)
9. Save settings and restart NPort server
10. It should be now possible to configure the device drivers interface with [IP]:4001. (example: 192.168.10.123:4001)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.test	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.model	CHOICE	SAVE SETUP	EnumRange (unknown ASC302LE ASC350L ASC300L-D ASC300LW ASC300C ASC300D-70 ASC300Ku1 ASC300Ku2 ASC300KuET ASC300X)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
frequency	FLOAT		DoubleRange (0.00 .. 0.00)
attn	FLOAT		DoubleRange (0.0 .. 60.0)
internal.v	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.a	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
internal.f	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.f1	TEXT	R/O	StringRange R/O
internal.f2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Lock alarm)

9.68 Auto-RollOff

The Auto-RollOff logical device adds an automatic roll-off selection to IRDs which lack from this feature or which do not behave properly with the roll-off set to AUTO.

The Auto-RollOff device is linked to an IRD's rollOff and symbolRate parameters, it provides an rollOff parameter which the GUI uses instead of the IRD's one. The device's rollOff parameter offers the same selections as the the parameter at the IRD, with an AUTO setting added.

If the rollOff parameter is set to one of the 20%, 25% or 35% settings, this value is passed to the IRD. If set to AUTO, the device computes and sets the rollOff setting from the symbol rate using these formulas:

1. Compute the occupied bandwidth *BWocc = symbolrate * 1.2*
2. From the occupied bandwidth evaluate the allocated bandwidth *BWalloc* by extending it to the next defined slot bandwidth in the list (see below).
3. Evaluate the roll off factor by comparing the allocated bandwidth to the symbolrate:*if symbolrate * 1.35 <= BWalloc then rolloff = 35% else if symbolrate * 1.25 <= BWalloc then rolloff = 25% else if symbolrate * 1.20 <= BWalloc then rolloff = 20%else raise fault* The evaluated roll of factor is commanded to the IRD, the same applies if the symbol rate changes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets the roll of factor
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Auto-RollOff device with the IRD device.

parameter	***description***
rollOffId	Enter the message identifier here of the IRD's roll off parameter, e.g. "IRD-3.rollOff".
outputId	Enter the message identifier here of the IRD's symbol rate parameter, e.g. "IRD-3.symbolRate".
bandwidthSteps	This parameter defines the stepping of allocated bandwidth as a comma separated list of values. Factory default is "3,4.5,6,9,12,18,24,27,36". For detail about specifying the list of allocated bandwidths see the paragraph below.

List of allocated bandwidths

The device provides a setup parameter to specify the list of allocated bandwidth value to use. When specifying this list, the following rules apply:

- The bandwidth values are floating point values, representing the bandwidth in Hz with the same magnification the IRD's symbol rate parameter uses. The IRD driver coming with the sat-nms software all specify the symbol rate in terms of Msym/sec, hence the bandwidth values are in MHz.
- The values must appear in ordered sequence, starting with the smallest bandwidth, ending with highest one. The last value also works as fallback for symbol rates which produce a higher occupied bandwidth.
- The list element must be separated by comma characters, spaces or other delimiters are not allowed.

Fault flags

The Auto-RollOff device provides one fault flag which is raised if in AUTO mode a symbol rate is recognized which occupies a bandwidth wider than the widest allocated slot, even with 20% roll off set. The flag remains active until a valid symbol rate is set at the IRD.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.rollOff	TEXT	R/O	StringRange R/O
info.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.bandwidthSteps	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Invalid symbol rate)
rollOff	CHOICE	NOPRESET	EnumRange (UNKNOWN)

9.69 AutomationDirect-Terminator

This page describes the device driver and the device window for the*AutomationDirect Terminator-IO* data IO unit. The device driver monitors and controls the digital inputs/outputs of various modules hosted by the Terminator-IO unit. The driver supports digital I/O modules only, analog modules cannot be monitored with this driver.

The *AutomationDirect-Terminator* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*AutomationDirect-Terminator* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the device driver as a low level driver to access the physical I/O ports.

Port addressing

The *AutomationDirect-Terminator* device driver provides 32 input and 32 output variables when each mirror 16 bits of the device's IO data space. This corresponds to the maximum number of IO modules which can be attached to the *AutomationDirect Terminator-IO* base device.

Without individual configuration of the device using the software coming with the *AutomationDirect Terminator-IO* , the device puts each single IO bit like beads on a chain, starting with the module next to the base unit at the lowest address. Input and output modules use different address spaces, hence if you put a 16 bit input and a 16 bit output module to the

base unit, the input bits appear in the driver's I01 variable, the output bits in Q01 - regardless if the input or the output module is mounted first.

As the device allocates addresses bitwise, there may be configurations where the inputs or outputs of a module are spread over two driver variables. This happens, if 8 and 16 bit modules are mixed.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Terminator-IO device acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the device in the device setup window , there must not be defined any other devices beside this module for this interface.
- The unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
I01	HEX	R/O	HexRange R/O (0 .. 0)
I02	HEX	R/O	HexRange R/O (0 .. 0)
I03	HEX	R/O	HexRange R/O (0 .. 0)
I04	HEX	R/O	HexRange R/O (0 .. 0)
I05	HEX	R/O	HexRange R/O (0 .. 0)
I06	HEX	R/O	HexRange R/O (0 .. 0)
I07	HEX	R/O	HexRange R/O (0 .. 0)
I08	HEX	R/O	HexRange R/O (0 .. 0)
I09	HEX	R/O	HexRange R/O (0 .. 0)
I10	HEX	R/O	HexRange R/O (0 .. 0)
I11	HEX	R/O	HexRange R/O (0 .. 0)
I12	HEX	R/O	HexRange R/O (0 .. 0)
I13	HEX	R/O	HexRange R/O (0 .. 0)
I14	HEX	R/O	HexRange R/O (0 .. 0)
I15	HEX	R/O	HexRange R/O (0 .. 0)
I16	HEX	R/O	HexRange R/O (0 .. 0)
I17	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
I18	HEX	R/O	HexRange R/O (0 .. 0)
I19	HEX	R/O	HexRange R/O (0 .. 0)
I20	HEX	R/O	HexRange R/O (0 .. 0)
I21	HEX	R/O	HexRange R/O (0 .. 0)
I22	HEX	R/O	HexRange R/O (0 .. 0)
I23	HEX	R/O	HexRange R/O (0 .. 0)
I24	HEX	R/O	HexRange R/O (0 .. 0)
I25	HEX	R/O	HexRange R/O (0 .. 0)
I26	HEX	R/O	HexRange R/O (0 .. 0)
I27	HEX	R/O	HexRange R/O (0 .. 0)
I28	HEX	R/O	HexRange R/O (0 .. 0)
I29	HEX	R/O	HexRange R/O (0 .. 0)
I30	HEX	R/O	HexRange R/O (0 .. 0)
I31	HEX	R/O	HexRange R/O (0 .. 0)
I32	HEX	R/O	HexRange R/O (0 .. 0)
Q01	HEX		HexRange (0 .. 0)
Q02	HEX		HexRange (0 .. 0)
Q03	HEX		HexRange (0 .. 0)
Q04	HEX		HexRange (0 .. 0)
Q05	HEX		HexRange (0 .. 0)
Q06	HEX		HexRange (0 .. 0)
Q07	HEX		HexRange (0 .. 0)
Q08	HEX		HexRange (0 .. 0)
Q09	HEX		HexRange (0 .. 0)
Q10	HEX		HexRange (0 .. 0)
Q11	HEX		HexRange (0 .. 0)
Q12	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
Q13	HEX		HexRange (0 .. 0)
Q14	HEX		HexRange (0 .. 0)
Q15	HEX		HexRange (0 .. 0)
Q16	HEX		HexRange (0 .. 0)
Q17	HEX		HexRange (0 .. 0)
Q18	HEX		HexRange (0 .. 0)
Q19	HEX		HexRange (0 .. 0)
Q20	HEX		HexRange (0 .. 0)
Q21	HEX		HexRange (0 .. 0)
Q22	HEX		HexRange (0 .. 0)
Q23	HEX		HexRange (0 .. 0)
Q24	HEX		HexRange (0 .. 0)
Q25	HEX		HexRange (0 .. 0)
Q26	HEX		HexRange (0 .. 0)
Q27	HEX		HexRange (0 .. 0)
Q28	HEX		HexRange (0 .. 0)
Q29	HEX		HexRange (0 .. 0)
Q30	HEX		HexRange (0 .. 0)
Q31	HEX		HexRange (0 .. 0)
Q32	HEX		HexRange (0 .. 0)
itick	INTEGER	R/O	IntegerRange R/O (0 .. 0)
otick	INTEGER	R/O	IntegerRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
i.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
i.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
i.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
T01	HEX	R/O	HexRange R/O (0 .. 0)
T02	HEX	R/O	HexRange R/O (0 .. 0)
T03	HEX	R/O	HexRange R/O (0 .. 0)
T04	HEX	R/O	HexRange R/O (0 .. 0)
T05	HEX	R/O	HexRange R/O (0 .. 0)
T06	HEX	R/O	HexRange R/O (0 .. 0)
T07	HEX	R/O	HexRange R/O (0 .. 0)
T08	HEX	R/O	HexRange R/O (0 .. 0)
T09	HEX	R/O	HexRange R/O (0 .. 0)
T10	HEX	R/O	HexRange R/O (0 .. 0)
T11	HEX	R/O	HexRange R/O (0 .. 0)
T12	HEX	R/O	HexRange R/O (0 .. 0)
T13	HEX	R/O	HexRange R/O (0 .. 0)
T14	HEX	R/O	HexRange R/O (0 .. 0)
T15	HEX	R/O	HexRange R/O (0 .. 0)
T16	HEX	R/O	HexRange R/O (0 .. 0)
T17	HEX	R/O	HexRange R/O (0 .. 0)
T18	HEX	R/O	HexRange R/O (0 .. 0)
T19	HEX	R/O	HexRange R/O (0 .. 0)
T20	HEX	R/O	HexRange R/O (0 .. 0)
T21	HEX	R/O	HexRange R/O (0 .. 0)
T22	HEX	R/O	HexRange R/O (0 .. 0)
T23	HEX	R/O	HexRange R/O (0 .. 0)
T24	HEX	R/O	HexRange R/O (0 .. 0)
T25	HEX	R/O	HexRange R/O (0 .. 0)
T26	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
T27	HEX	R/O	HexRange R/O (0 .. 0)
T28	HEX	R/O	HexRange R/O (0 .. 0)
T29	HEX	R/O	HexRange R/O (0 .. 0)
T30	HEX	R/O	HexRange R/O (0 .. 0)
T31	HEX	R/O	HexRange R/O (0 .. 0)
T32	HEX	R/O	HexRange R/O (0 .. 0)

9.70 AvL-ITC-ACU

This page describes the device driver and the device window for the AvL ITC antenna control unit. Actually the device driver supports TLE tracking and antenna pointing, other tracking modes are not yet supported.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and permits to set new pointing angles.
- --- The second parameter page displays / configures the parameters for TLE tracking.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	For the AvL ACU in this field the IP4 address (with optional port) followed by user name and password must be entered. Example: `10.10.1.1:4321 Integrator Int123`

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
true	BOOLEAN	R/O	BooleanRange R/O (true false)
false	BOOLEAN	R/O	BooleanRange R/O (true false)
mustInitAzEl	TEXT	R/O	StringRange R/O
telemetryRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
backgroundCommands	TEXT	R/O	StringRange R/O
activeCommand	TEXT	R/O	StringRange R/O
config.longitude	FLOAT	SETUP	DoubleRange (-180.000 .. 360.000)
config.latitude	FLOAT	SETUP	DoubleRange (-90.000 .. 90.000)
config.altitude	FLOAT	SETUP	DoubleRange (0 .. 16000)
azimuth	FLOAT		DoubleRange (0.0000 .. 0.0000)
elevation	FLOAT		DoubleRange (0.0000 .. 0.0000)
polarization	FLOAT		DoubleRange DIS (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
state.el	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.pl	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
target	TEXT	SAVE	StringRange
tle1	TEXT	SAVE	StringRange
tle2	TEXT	SAVE	StringRange
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
updateTarget	TEXT	NOPRESET	StringRange
moveToPosition	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
startTLE	TEXT	NOPRESET	StringRange
gsa.target	TEXT		StringRange
gsa.dtrfreq	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.tle1	TEXT		StringRange
gsa.tle2	TEXT		StringRange
gsa.startTrack	TEXT	NOPRESET	StringRange
gsa.stopTrack	TEXT	NOPRESET	StringRange
gsa.doPark	TEXT		StringRange
gsa.azPark	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.elPark	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (X Motor Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (X Motor Warning)

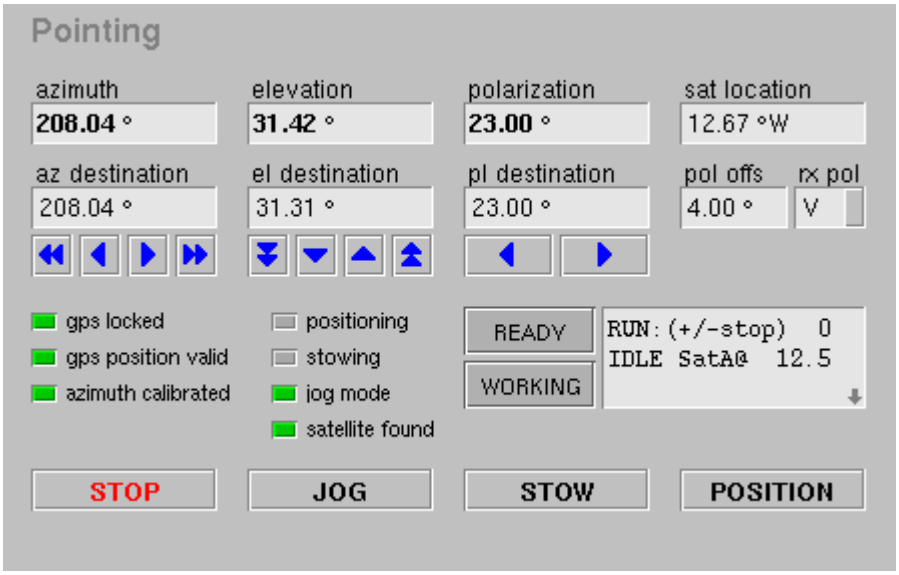
name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Y Motor Alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Y Motor Warning)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Beacon Receiver)

9.71 AvL-TracStar

This page describes the device driver and the device window for the AvL TracStar car mounted antenna controller.

Basic Operation

The following short description of operating the TracStar with the sat-nms software presumes you are already used to operate the TracStar using it's handheld terminal or indoor unit. If in doubt, consult the TracStar user manual.



The buttons at the bottom of the TracStar device window control the basic functions of the device:

- **STOP** --- Stops the antenna motors.
- **JOG** --- Turns the TracStar to jog mode. In jog mode you may command the antenna pointing angles directly or increments them using the arrow buttons. The 'rx pol' setting directly controls the receive polarization plane in jog mode and setting the 'sat location' makes the device driver calculate and command the pointing angles for the given satellite orbit position
- **STOW** --- Moves the antenna to stow position. Unfortunately the TracStar does not deliver a reliable 'STOWED' indication to the VLC, hence you should inspect visually if the

antenna is stowed properly before you move the car.
- **POSITION** --- Makes the TracStar elevate the antenna and search the target satellite which has been defined at the *Satellites* page. In most cases this starts the full search procedure including compass calibration, reference satellite identification and peaking to the target satellite. When the TracStar successfully peaked the target satellite, the driver switches to jog mode and applies the polarization offset.

To point the TracStar antenna to a satellite, you will typically perform the following steps:

1. At the *Satellites* page select two reference satellites.
2. Enter the parameters of the target satellite at the top of this page. Alternatively you may want to recall a device preset which contains these selections.
3. At the *Main* page of the device window click 'POSITION'. The TracStar now wil start it's full positioning cycle, after typically 2 or 3 minutes the antenna os positioned to the satellite.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing.
- --- This page shows the readings of the GPS receiver, the inclinometer and the receive levels.
- --- This page contains the identification parameters of the target satellite and the reference satellites.
- --- This page gives access to a number of parameters which tell the TrackStar how to reach and identify satellites. You should only change these values if you are already used to tweak the TracStar's behaviour ba modifying it's 'LOCKVAL' parameter.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
freqMode	Main operation mode of the TracStar
jogspeed	The motor speed to be used when in jog mode. 2 or 3 degrees/sec are practicable values.

parameter	***description***
lnbGain	Receiver sensitivity. The default value is 100. May be adjusted if the level reading of strong satellites reaches the '2999'.
lnbOffset	A second receiver sensitivity parameter.
loFreqLow	The 11GHz band LO frequency. See the chapter 'Receive Band Switching' below for details.
loFreqHigh	The 12GHz band LO frequency.
lnbSwitchId	The parameter ID to send band switching commands to. Example: 'LNB-PSU.22kHz'
lnbSwitchLow	The command to send to the parameter ID above to switch to 11GHz band.
lnbSwitchHigh	The command to send to the parameter ID above to switch to 11GHz band.
scanLStart	Scan start frequency for the 11GHz band.
scanLStop	Scan stop frequency for the 11GHz band.
scanHStart	Scan start frequency for the 12GHz band.
scanHStop	Scan stop frequency for the 12GHz band.
rxBand	Selects 11/12GHz band.
Serial	Remember to set the line parameters for the serial interface connected to the TracStar. The TracStar expects *9600,E,8,1* .

Receive Band Switching

The TracStar itself does not support a switch between two receive bands as commonly used in Europe. The sat-nms device driver add (limited) support for this. For this you require a LNB power supply which can be controlled by the VLC in order to switch the receive band. With the configuration setting above, configure the message ID and parameter values necessary to switch the band. Then, using the *rxBand* configuration parameter (it is available at the *Satellites* page as well), you may select the frequency band as required.

Any time you command the TracStar through the sat-nms software to position the antenna to the target satellite or to perform a sky scan, the driver first sets the receive band as selected and commands the LO frequency and scan start / stop frequencies configured for the selected receive band. satellite or to

As the TracStar itself is not aware of this extension, the following limitations apply:

- No plausibility check between the selected receive / channel frequencies and the selected

band is done. The TracStar probably will not find the satellite if you select the high band and supply low abd frequencies.
- The frequencies used with the target satellite and the reference satellites must be in the same frequency band because the TracStar is not capable to switch the band while positioning the antenna.

The wildcard frequency 10799 MHz applies to both frequency bands, you may use it with the high frequency as well if you want to make the TracStar search *any* channel at this satellite.

Polarization Plane Handling

The TracStar provides a polarization plane parameter which turns the feed by 90 degree if set to vertical. The polarization angle displayed (and commanded) is meant as an offset to the selected polarization plane.

The polarization plane parameter only actively changes the feed angle if the TracStar is in jog mode. In IDLE / STOPPED mode the polarization setting is memorized, the setting is postponed until a POSITION command is given.

When positioning to a satellite, the TracStar applies a polarization offset which is calculated from the antenna's geodetic location and the satellite's orbit position. It does not know about the particular skew offset of the satellite. The sat-nms driver adds this feature to the satellite positioning procedure:

The parameter 'pol offs' refers to the skew offset of the target satellite. The parameter gets saved with each particular device (=satellite) preset. After the TracStar has successfully positioned the antenna to the target satellite, the driver switches to jog mode and applies the skew offset. This way you may use stored device presets as an instant setup for already polarization-optimized pointing.

ATTENTION: The TracStar does not honour the polarization plane parameter when performing a sky scan. The channel frequencies of the reference satellites always are searched at horizontal polarization. If you want to manage a list of frequencies found at vertical polarization, you have to search these frequencies manually, e.g. with a spectrum analyzer.

Device Presets

The device driver stores complete setups, including the target satellite parameters, all reference satellites and the selection of those. A device preset of the AvL-TracStar device is intended to act as a data file of all information required to point the antenna to a certain satellite. When the preset is recalled, the satellite parameters get programmed to the TracStar, but no positioning is commanded. Due to this fact, you may recall presets in order to view / edit / copy them without moving the antenna. Once you have recalled the preset with the satellite parameters you are looking for, click 'POSITION' to make the TracStar point the antenna to the selected satellite.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.faultcode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.oper	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.pstat	TEXT	R/O	StringRange R/O
internal.line1	TEXT	R/O	StringRange R/O
internal.line2	TEXT	R/O	StringRange R/O
internal.lofreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.scan2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.scan1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lok	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.freqMode	CHOICE	SETUP	EnumRange (USA USA-DVB EUROPE MIDEAST)
config.jogspeed	INTEGER	SAVE SETUP	IntegerRange (1 .. 5)
config.lnbGain	INTEGER	SETUP	IntegerRange (50 .. 300)

name	*type*	*flags*	*range*
config.lnbOffset	INTEGER	SETUP	IntegerRange (0 .. 900)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lnbSwitchId	TEXT	SAVE SETUP	StringRange
config.lnbSwitchLow	TEXT	SAVE SETUP	StringRange
config.lnbSwitchHigh	TEXT	SAVE SETUP	StringRange
config.scanLStart	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.scanLStop	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.scanHStart	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.scanHStop	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rxBand	CHOICE	SAVE SETUP	EnumRange (LOW HIGH)
config.presetDefault	TEXT	R/O	StringRange R/O
azimuth	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
elevation	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
polarization	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
orbit	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
rxpolplane	CHOICE	SAVE	EnumRange (H V)
jogpolplane	CHOICE	NOPRESET	EnumRange (H V)
step	TEXT	NOPRESET	StringRange
jog	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
deploy	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
skyscan	TEXT	NOPRESET	StringRange
refresh	TEXT	NOPRESET	StringRange
selref	TEXT	NOPRESET	StringRange
evalPolOffset	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.aped	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.aoffs	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.nick	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.roll	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.longitude	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.latitude	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.magdev	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.height	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.gpsok	BOOLEAN	R/O	BooleanRange R/O (true false)
state.gpslock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.azcal	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ready	BOOLEAN	R/O	BooleanRange R/O (true false)
state.working	BOOLEAN	R/O	BooleanRange R/O (true false)
state.display	TEXT	R/O	StringRange R/O
state.mode	TEXT	R/O	StringRange R/O
state.positioning	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stowing	BOOLEAN	R/O	BooleanRange R/O (true false)
state.jogmode	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.satfound	BOOLEAN	R/O	BooleanRange R/O (true false)
state.agc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.snr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.rxfreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sat.name	TEXT	SAVE	StringRange
sat.location	FLOAT		DoubleRange (0.0 .. 0.0)
sat.f1	INTEGER		IntegerRange (0 .. 0)
sat.f2	INTEGER		IntegerRange (0 .. 0)
sat.srate	INTEGER		IntegerRange (0 .. 0)
sat.auto	TEXT	NOPRESET	StringRange
sat.ref1	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12)
sat.ref2	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12)
sat.poloffs.h	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
sat.poloffs.v	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
pointMode	CHOICE		EnumRange (DIRECT REFERENCE)
satelliteScan	CHOICE		EnumRange (STRENGTH FREQUENCY)
lockCondition	CHOICE		EnumRange (AGC AGC & SNR SNR ONLY)
refFCondition	CHOICE		EnumRange (NONE SLOW F1 OR F2 F1 AND F2)
datFCondition	CHOICE		EnumRange (NONE F1 OR F2 F1 AND F2)
ref.1.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.1.f1	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ref.1.f2	INTEGER		IntegerRange (0 .. 0)
ref.1.srate	INTEGER		IntegerRange (0 .. 0)
ref.1.mode	CHOICE		EnumRange (SS FREQ)
ref.1.qual	INTEGER		IntegerRange (0 .. 0)
ref.1.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.1.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.1.auto	TEXT	NOPRESET	StringRange
ref.1.rxpol	TEXT	SAVE	StringRange
ref.2.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.2.f1	INTEGER		IntegerRange (0 .. 0)
ref.2.f2	INTEGER		IntegerRange (0 .. 0)
ref.2.srate	INTEGER		IntegerRange (0 .. 0)
ref.2.mode	CHOICE		EnumRange (SS FREQ)
ref.2.qual	INTEGER		IntegerRange (0 .. 0)
ref.2.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.2.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.2.auto	TEXT	NOPRESET	StringRange
ref.2.rxpol	TEXT	SAVE	StringRange
ref.3.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.3.f1	INTEGER		IntegerRange (0 .. 0)
ref.3.f2	INTEGER		IntegerRange (0 .. 0)
ref.3.srate	INTEGER		IntegerRange (0 .. 0)
ref.3.mode	CHOICE		EnumRange (SS FREQ)
ref.3.qual	INTEGER		IntegerRange (0 .. 0)
ref.3.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.3.available	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ref.3.auto	TEXT	NOPRESET	StringRange
ref.3.rxpol	TEXT	SAVE	StringRange
ref.4.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.4.f1	INTEGER		IntegerRange (0 .. 0)
ref.4.f2	INTEGER		IntegerRange (0 .. 0)
ref.4.srate	INTEGER		IntegerRange (0 .. 0)
ref.4.mode	CHOICE		EnumRange (SS FREQ)
ref.4.qual	INTEGER		IntegerRange (0 .. 0)
ref.4.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.4.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.4.auto	TEXT	NOPRESET	StringRange
ref.4.rxpol	TEXT	SAVE	StringRange
ref.5.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.5.f1	INTEGER		IntegerRange (0 .. 0)
ref.5.f2	INTEGER		IntegerRange (0 .. 0)
ref.5.srate	INTEGER		IntegerRange (0 .. 0)
ref.5.mode	CHOICE		EnumRange (SS FREQ)
ref.5.qual	INTEGER		IntegerRange (0 .. 0)
ref.5.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.5.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.5.auto	TEXT	NOPRESET	StringRange
ref.5.rxpol	TEXT	SAVE	StringRange
ref.6.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.6.f1	INTEGER		IntegerRange (0 .. 0)
ref.6.f2	INTEGER		IntegerRange (0 .. 0)
ref.6.srate	INTEGER		IntegerRange (0 .. 0)
ref.6.mode	CHOICE		EnumRange (SS FREQ)

name	*type*	*flags*	*range*
ref.6.qual	INTEGER		IntegerRange (0 .. 0)
ref.6.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.6.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.6.auto	TEXT	NOPRESET	StringRange
ref.6.rxpol	TEXT	SAVE	StringRange
ref.7.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.7.f1	INTEGER		IntegerRange (0 .. 0)
ref.7.f2	INTEGER		IntegerRange (0 .. 0)
ref.7.srate	INTEGER		IntegerRange (0 .. 0)
ref.7.mode	CHOICE		EnumRange (SS FREQ)
ref.7.qual	INTEGER		IntegerRange (0 .. 0)
ref.7.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.7.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.7.auto	TEXT	NOPRESET	StringRange
ref.7.rxpol	TEXT	SAVE	StringRange
ref.8.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.8.f1	INTEGER		IntegerRange (0 .. 0)
ref.8.f2	INTEGER		IntegerRange (0 .. 0)
ref.8.srate	INTEGER		IntegerRange (0 .. 0)
ref.8.mode	CHOICE		EnumRange (SS FREQ)
ref.8.qual	INTEGER		IntegerRange (0 .. 0)
ref.8.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.8.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.8.auto	TEXT	NOPRESET	StringRange
ref.8.rxpol	TEXT	SAVE	StringRange
ref.9.location	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
ref.9.f1	INTEGER		IntegerRange (0 .. 0)
ref.9.f2	INTEGER		IntegerRange (0 .. 0)
ref.9.srate	INTEGER		IntegerRange (0 .. 0)
ref.9.mode	CHOICE		EnumRange (SS FREQ)
ref.9.qual	INTEGER		IntegerRange (0 .. 0)
ref.9.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.9.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.9.auto	TEXT	NOPRESET	StringRange
ref.9.rxpol	TEXT	SAVE	StringRange
ref.10.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.10.f1	INTEGER		IntegerRange (0 .. 0)
ref.10.f2	INTEGER		IntegerRange (0 .. 0)
ref.10.srate	INTEGER		IntegerRange (0 .. 0)
ref.10.mode	CHOICE		EnumRange (SS FREQ)
ref.10.qual	INTEGER		IntegerRange (0 .. 0)
ref.10.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.10.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.10.auto	TEXT	NOPRESET	StringRange
ref.10.rxpol	TEXT	SAVE	StringRange
ref.11.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.11.f1	INTEGER		IntegerRange (0 .. 0)
ref.11.f2	INTEGER		IntegerRange (0 .. 0)
ref.11.srate	INTEGER		IntegerRange (0 .. 0)
ref.11.mode	CHOICE		EnumRange (SS FREQ)
ref.11.qual	INTEGER		IntegerRange (0 .. 0)
ref.11.used	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ref.11.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.11.auto	TEXT	NOPRESET	StringRange
ref.11.rxpol	TEXT	SAVE	StringRange
ref.12.location	FLOAT		DoubleRange (0.0 .. 0.0)
ref.12.f1	INTEGER		IntegerRange (0 .. 0)
ref.12.f2	INTEGER		IntegerRange (0 .. 0)
ref.12.srate	INTEGER		IntegerRange (0 .. 0)
ref.12.mode	CHOICE		EnumRange (SS FREQ)
ref.12.qual	INTEGER		IntegerRange (0 .. 0)
ref.12.used	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.12.available	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.12.auto	TEXT	NOPRESET	StringRange
ref.12.rxpol	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Azimuth)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Polarization)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Elevation)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Base angle limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Reference satellite)
faults.06	ALARM	R/O	AlarmFlagRange R/O (XPonder signal)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Compass)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Other fault)

9.72 AzElCollector

The AzElcollector logical device provides a string containing the azimuth, elevation and polarization angle of an antenna, followed by the beacon level and frequency and (if applicable) the az/el error values for a monopulse tracking. This string is provided as the 'poslog' variable of the AzElCollector device, it has the following format: ***YYYY-MM-DD***

HH:MM:SS;a.aaa;e.eee;p.dotppp;l.ll;f.ffffff;x.xxx;y.yyy;ttt with

- *a.aaa* = azimuth angle [°]
- *e.eee* = elevation angle [°]
- *p.ppp* = polarization angle [°]
- *l.ll* = beacon level [dBm]
- *f.ffffff* = beacon frequency [MHz]
- *x.xxx* = monopulse azimuth error
- *y.yyy* = monopulse elevation error
- *ttt* = tracking state, as reported from the antenna

Timestamp and the antenna pointing values are created by the antenna device driver, so the timestamp exactly states the time when these values have been read from the device. The precision of the values is defined by the variable definition of each value in the device which originally reads the value from a device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the 'poslog' string for information.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the AzElCollector device to the devices it monitors.

parameter	***description***
acuPoslogId	The parameter ID of the 'poslog' variable provided by the antenna device driver. This typically is something like 'ANTENNA.info.poslog'.
beaconLevelId	The parameter ID of the variable containing the actual beacon level.
beaconFrequencyId	The parameter ID of the variable containing the beacon frequency.
monopulseAzErrorId	The parameter ID of the variable showing the actual monopulse azimuth error
monopulseElErrorId	The parameter ID of the variable showing the actual monopulse elevation error
trackingStateId	The parameter ID of the variable showing the actual tracking state.

All IDs except the first one may be left empty, the resulting poslog string then contains '0' for this value if this is a numerical one or an empty string if the missing parameter id refers to the tracking state.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.acuPoslogId	TEXT	SAVE SETUP	StringRange
config.beaconLevelId	TEXT	SAVE SETUP	StringRange
config.beaconFrequencyId	TEXT	SAVE SETUP	StringRange
config.monopulseAzErrorId	TEXT	SAVE SETUP	StringRange
config.monopulseElErrorId	TEXT	SAVE SETUP	StringRange
config.trackingStateId	TEXT	SAVE SETUP	StringRange
poslog	TEXT	R/O	StringRange R/O

9.73 BISS-KeySelect-CDB-DBonly

The BISS-KeySelect-CDB logical device lets you comfortably set BISS encryption keys from a central database. This is a special version of the device, which provides only access to the database.

"CDB" in the driver name stands for "Central Data Base", underlining that all BISS-KeySelect-CDB instances use one central BISS key database located at the NMS server. The BISS-KeySelect-CDB device cannot be run in a plain M&C environment.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The BISS key database editor page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (database edit)

With the second page of the device window of the BISS-KeySelect-CDB device BISS key definitions are edited, copied or deleted. All BISS-KeySelect-CDB devices in a sat-nms NMS share one common BISS key database located at the NMS server. The VLC containing the BISS-KeySelect-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Editing a BISS key definition does not change the encryption key in the associated device.

- Add/Change a key definition:
  1. Select the encryption key to modify
  2. Edit the key
  3. Click 'SAVE'Please note that if you change the textual description of the key, this will store a new key definition with the modified data, the old definition is kept unchanged in the database.The textual description of the key may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the key gets stored.
- Delete a key definition:
  1. Select the key definition to delete
  2. Click 'DELETE'

Local database copy

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the BISS-KeySelect-CDB functionality even in this backup case, the VLC maintains a local copy of the BISS key database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the keys in this local backup copy, but you can apply the stored keys to a device.

There are two ways to update the local copy of the BISS key database at the VLC:

1. At the "Database" page of the device window of a BISS-KeySelect-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.bisskeys.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

Configuration parameters

Just leave all configuration parameters empty.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.bissmodeId	TEXT	SAVE SETUP	StringRange
config.bisskeyId	TEXT	SAVE SETUP	StringRange
config.bisskeyLoId	TEXT	SAVE SETUP	StringRange
config.bisskeyHiId	TEXT	SAVE SETUP	StringRange
key	CHOICE		EnumRange (@@@bisskey12)
bissMode	CHOICE		EnumRange (UNKNOWN)
edit.key	CHOICE		EnumRange (@@@bisskey12)
edit.hexKey	TEXT		StringRange
edit.description	TEXT		StringRange
edit.bissMode	CHOICE		EnumRange (BISS-1 BISS-E)
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O

9.74 BISS-KeySelect-CDB

The BISS-KeySelect-CDB logical device lets you comfortably set BISS encryption keys from a central database. You may store the hexadecimal keys with a user defined description in the database and retrieve them by this descriptive text.

"CDB" in the driver name stands for "Central Data Base", underlining that all BISS-KeySelect-CDB instances use one central BISS key database located at the NMS server. The BISS-KeySelect-CDB device cannot be run in a plain M&C environment.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (BISS key select) page for the device.
- --- The BISS key database editor page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (BISS key select)

The device simply provides a single entry field as an interface to the operator. From this entry field you can select any key stored in the database.

If you enter a 12 or 16 digit hexadecimal number to the field, the device treats this as a BISS key to be forwarded without using the database. This way you easily can use encryption keys which are not stored in the database.

Most IRD devices do not allow to read back the BISS encryption key. The BISS-KeySelect-CDB device defangs this restriction by leaving the last key applied in the entry field. You should be aware, that there may be situations where the displayed key is not that one actually active in the IRD.

Operation (database edit)

With the second page of the device window of the BISS-KeySelect-CDB device BISS key definitions are edited, copied or deleted. All BISS-KeySelect-CDB devices in a sat-nms NMS share one common BISS key database located at the NMS server. The VLC containing the BISS-KeySelect-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Editing a BISS key definition does not change the encryption key in the associated device.

- Add/Change a key definition:
  1. Select the encryption key to modify
  2. Edit the key
  3. Click 'SAVE'Please note that if you change the textual description of the key, this will store a new key definition with the modified data, the old definition is kept unchanged in the database.The textual description of the key may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the key gets stored.
- Delete a key definition:
  1. Select the key definition to delete
  2. Click 'DELETE'

Local database copy

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the BISS-KeySelect-CDB functionality even in this backup case, the VLC maintains a local copy of the BISS key database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the keys in this local backup copy, but you can apply the stored keys to a device.

There are two ways to update the local copy of the BISS key database at the VLC:

1. At the "Database" page of the device window of a BISS-KeySelect-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add two lines **cdb.bisskey12.syncInterval=#.# cdb.bisskey16.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the BISS-KeySelect-CDB device with the equipment.

parameter	***description***
bisskeyId	The message ID to set the encryption key at the device. If the device requires the key to be split in a lower and an upper part, leave this field empty and use the two fields below.
bisskeyLoId	The message ID to set the lower part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.
bisskeyHiId	The message ID to set the higher part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.bissmodeId	TEXT	SAVE SETUP	StringRange
config.bisskeyId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.bisskeyLoId	TEXT	SAVE SETUP	StringRange
config.bisskeyHiId	TEXT	SAVE SETUP	StringRange
key	CHOICE		EnumRange (@@@bisskey12)
bissMode	CHOICE		EnumRange (UNKNOWN)
edit.key	CHOICE		EnumRange (@@@bisskey12)
edit.hexKey	TEXT		StringRange
edit.description	TEXT		StringRange
edit.bissMode	CHOICE		EnumRange (BISS-1 BISS-E)
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O

9.75 BISS-KeySelect

The BISS-KeySelect logical device lets you comfortably set BISS encryption keys from a central database. You may store the hexadecimal keys with a user defined description in the database and retrieve them by this descriptive text.

BISS-KeySelect reads its key definitions from the satellite databse (SQL database). It replaces the the older BISS-KeySelect-CDB device. **Device window pages**

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (BISS key select) page for the device.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

To set a BISS key at the device, first select the BISS mode (BISS-E, BISS-1, ...). Then select the

key by its name from the list of keys and press enter.

Alternatively you may enter a 12 or 16 digit hexadecimal number to the field, the device treats this as a BISS key to be forwarded without using the database. This way you easily can use encryption keys which are not stored in the database.

Most IRD devices do not allow to read back the BISS encryption key. The BISS-KeySelect-CDB device defangs this restriction by leaving the last key applied in the entry field. You should be aware, that there may be situations where the displayed key is not that one actually active in the IRD.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the BISS-KeySelect device with the equipment.

parameter	***description***
bissmodeId	The message ID to set the BISS mode at the device.
bisskeyId	The message ID to set the encryption key at the device. If the device requires the key to be split in a lower and an upper part, leave this field empty and use the two fields below.
bisskeyLoId	The message ID to set the lower part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.
bisskeyHiId	The message ID to set the higher part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.

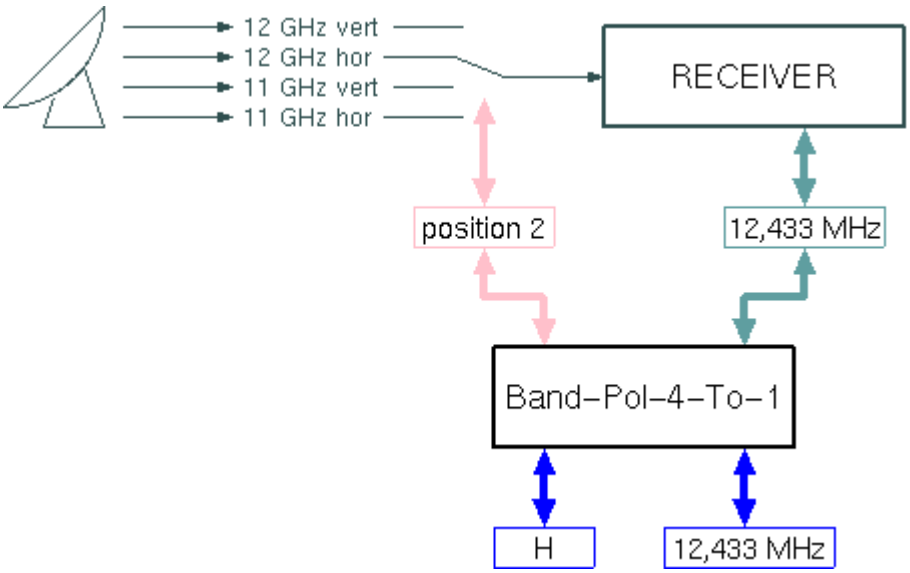
Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.bissmodeId	TEXT	SAVE SETUP	StringRange
config.bisskeyId	TEXT	SAVE SETUP	StringRange
config.bisskeyLoId	TEXT	SAVE SETUP	StringRange
config.bisskeyHiId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
key	CHOICE		EnumRange (unknown)
bissMode	CHOICE		EnumRange (unknown)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O

9.76 Band-Pol-4-To-1

The Band-Pol-4-To-1 logical device controls a 4-to-1 switch which typically selects a receiver input signal from two polarization panes and two frequency bands. The switch either may be part of the receiver or an external switch.



In a typical configuration the Band-Pol-4-To-1 device is placed between the Link device and the receiver. The Link device sets the receive polarization and frequency at the Band-Pol-4-To-1 device. The latter selects the proper input signal and forwards the frequency setting to the receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control page for the device. You may set the frequency and polarization

manually at this page.

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Frequency-Split device with the devices that will perform the frequency setting.

parameter	***description***
frequencyId	Enter the message identifier here of the parameter which shall receive the frequency setting, e.g. "IRD-1.frequency". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "IRD-1.select". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
freqThresh	The frequency threshold which separates the "low" band from the "hi" band. With Ku Band receivers you usually will set this to 11.800 MHz.
lowVertPos	The switch position to be set to select the low band, vertical polarization signal.
lowHorPos	The switch position to be set to select the low band, horizontal polarization signal.
hiVertPos	The switch position to be set to select the high band, vertical polarization signal.
hiHorPos	The switch position to be set to select the high band, horizontal polarization signal.

To use the Band-Pol-4-To-1 logical device as a polarization select control only, set the freqThreshold to the lowest possible value and use the high band switch positions.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.switchId	TEXT	SAVE SETUP	StringRange
config.freqThresh	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.lowVertPos	TEXT	SAVE SETUP	StringRange
config.hiVertPos	TEXT	SAVE SETUP	StringRange
config.lowHorPos	TEXT	SAVE SETUP	StringRange
config.hiHorPos	TEXT	SAVE SETUP	StringRange
position	TEXT	R/O	StringRange R/O
polarization	CHOICE		EnumRange (X Y)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)

9.77 Barco-RE4220

This page describes the device driver and the device window for the Barco RE4220 MPEG Video Encoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- MPEG output parameters.
- --- Video parameters.
- --- Audio parameters.
- --- Auxiliary data parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

parameter	*description*

parameter	***description***
address	Contains the address of the encoder. The address consists of two numbers: DeviceAddress and RemoteAddress. The DeviceAddress describes the Product type and is *82* for the Barco encoder. The RemoteAddress identifies a single Device. Leave this field empty to use the autodetect feature of the driver.
audioChannel[1..4]	Selects the audio channel
audioModeAdaption[1..4]	Enables auto adaption to digital input. The currents settings of the encoder are read at start-up but the setting of the encoder will only change if the appropriate audio mode of a channel is set.
Serial	Remember to set the line parameters for the serial interface connected to the encoder. The Barco encoder expects *19200,N,8,1* .

Communication protocol The Barco Encoder device driver requires the *Scientific-Atlanta RCDS* protocol.

Remote interfaces

The Front Panel interface of the Barco Encoder provides an RS232 interface. It expects *19200,N,8,1* and an complete 1:1 cable (all 9 pins). After pressing*RETURN* the Encoder shows a prompt, but the input is not echod to the screen. Try *?* to get the device type and version of the encoder. To acquire the RemoteAddress use *SYS:RCDS REM:ADDR?* . To set the RemoteAddress manually use *SYS:RCDS REM:ADDR=n* where n is the new address.

To connect the Barco encoder to the sat-nms system use the RS485 male DSUB-9 connector (labeled as AUX 2) at the back side of the encoder.

- --- BarcoVLC
- TxB --- 24
- RxB --- 31
- TxA --- 43
- RxA --- 52
- GND --- 75

Remarks

- The Barco encoder supports up to 3 video channels but this driver supports only the first video channel. Please disable channel 2 and 3 before using the sat-nms system to control the encoder.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.audioChannel	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
config.audioModeAdaptation1	CHOICE	SAVE SETUP	EnumRange (AUTO OFF)
config.audioModeAdaptation2	CHOICE	SAVE SETUP	EnumRange (AUTO OFF)
config.audioModeAdaptation3	CHOICE	SAVE SETUP	EnumRange (AUTO OFF)
config.audioModeAdaptation4	CHOICE	SAVE SETUP	EnumRange (AUTO OFF)
reset	TEXT	NOPRESET	StringRange
info.identification	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
output.datarate	INTEGER		IntegerRange (1000000 .. 96000000)
output.framing	CHOICE		EnumRange (OFF G704 G832 G751 DVB34 G804 G752 DVB45)
output.asimode	CHOICE		EnumRange (PACKET BURST BYTE)
output.delay	INTEGER		IntegerRange (0 .. 2000)

name	*type*	*flags*	*range*
output.clksource	CHOICE		EnumRange (INTernal EXTernal TRACK)
output.profile	CHOICE		EnumRange (4:2:0 4:2:2)
output.packetsize	CHOICE		EnumRange (188RSOFF 204RSON 204RSOFF)
output.service	TEXT		StringRange
output.program	INTEGER		IntegerRange (1 .. 65525)
output.progmode	CHOICE		EnumRange (DISable AUTO MANual)
output.pmtpid	INTEGER		IntegerRange (32 .. 8187)
video.source	CHOICE	R/O	EnumRange R/O (Composite SDI)
video.format	CHOICE		EnumRange (A COMP:PAL COMP:NTSC COMP:PALPLUS SDI:525 SDI:625 COMP:AUTO SDI:AUTO)
video.datarate	INTEGER		IntegerRange (1500000 .. 50000000)
video.resolution	CHOICE		EnumRange (720 704 360)
video.pid	INTEGER		IntegerRange (32 .. 8187)
audio.1.datarate	CHOICE		EnumRange DIS (64 kbps 128 kbps 192 kbps 256 kbps 384 kbps)
audio.1.sampling	CHOICE		EnumRange DIS (32.0 kHz 44.1 kHz 48.0 kHz)
audio.1.source	CHOICE		EnumRange DIS (ANAlog DIGItal EMBedded)
audio.1.mode	CHOICE		EnumRange DIS (STEREO JOINT DUAL MONO)
audio.1.pid	INTEGER		IntegerRange DIS (32 .. 8187)

name	*type*	*flags*	*range*
audio.2.datarate	CHOICE		EnumRange DIS (64 kbps 128 kbps 192 kbps 256 kbps 384 kbps)
audio.2.sampling	CHOICE		EnumRange DIS (32.0 kHz 44.1 kHz 48.0 kHz)
audio.2.source	CHOICE		EnumRange DIS (ANAlog DIGItal EMBedded)
audio.2.mode	CHOICE		EnumRange DIS (STEREO JOINT DUAL MONO)
audio.2.pid	INTEGER		IntegerRange DIS (32 .. 8187)
audio.3.datarate	CHOICE		EnumRange DIS (64 kbps 128 kbps 192 kbps 256 kbps 384 kbps)
audio.3.sampling	CHOICE		EnumRange DIS (32.0 kHz 44.1 kHz 48.0 kHz)
audio.3.source	CHOICE		EnumRange DIS (ANAlog DIGItal EMBedded)
audio.3.mode	CHOICE		EnumRange DIS (STEREO JOINT DUAL MONO)
audio.3.pid	INTEGER		IntegerRange DIS (32 .. 8187)
audio.4.datarate	CHOICE		EnumRange DIS (64 kbps 128 kbps 192 kbps 256 kbps 384 kbps)
audio.4.sampling	CHOICE		EnumRange DIS (32.0 kHz 44.1 kHz 48.0 kHz)
audio.4.source	CHOICE		EnumRange DIS (ANAlog DIGItal EMBedded)
audio.4.mode	CHOICE		EnumRange DIS (STEREO JOINT DUAL MONO)
audio.4.pid	INTEGER		IntegerRange DIS (32 .. 8187)
auxdata.datarate	CHOICE		EnumRange (600 bps 1200 bps 2400 bps 4800 bps 9600 bps 19200 bps 38400bps)

name	*type*	*flags*	*range*
auxdata.on	CHOICE		EnumRange (OFF ON)
auxdata.pid	INTEGER		IntegerRange (32 .. 8187)
teletext.pid	INTEGER		IntegerRange (32 .. 8187)
internal.status	TEXT	R/O	StringRange R/O
internal.faults	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dummy	TEXT		StringRange
internal.error	INTEGER		IntegerRange (0 .. 0)
internal.t1	TEXT		StringRange
internal.t2	TEXT		StringRange
internal.t3	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Comp. Video input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (SDI Video input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video codec)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Network input)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Network input event)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Network output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Emb. Audio input)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Dig. Audio input)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module config)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (System config)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Power Supply)

9.78 Bard-MC4002-AC

This page describes the read only device driver and the device window for the Bard MC4002 Solid State Dual Unit Lead/Lag Controller with two HVAC units via SNMP.

It was tested with FW revision: 2.15

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters and measurement readings.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple

lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.site	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model.1	TEXT	R/O	StringRange R/O
info.model.2	TEXT	R/O	StringRange R/O
info.serial.1	TEXT	R/O	StringRange R/O
info.serial.2	TEXT	R/O	StringRange R/O
info.alarmBoard	CHOICE	R/O	EnumRange R/O (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.sensor.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sensor.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp.localRoom	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.sensor.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.sensor.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.localRoom.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.temp.sensor.1.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.sensor.2.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.cool	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.heat	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.highTemp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.highTemp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.lowTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.cool.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
setpoint.heat.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
setpoint.highTemp.1.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
setpoint.highTemp.2.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
setpoint.lowTemp.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.generator	CHOICE	R/O	EnumRange R/O (OFF ON)
state.comfortMode	CHOICE	R/O	EnumRange R/O (DISABLED ENABLED)
state.controller	CHOICE	R/O	EnumRange R/O (UNLOCKED LOCKED)
state.hvac.1	CHOICE	R/O	EnumRange R/O (LEAD LAG)
state.hvac.2	CHOICE	R/O	EnumRange R/O (LEAD LAG)
state.fan.1	CHOICE	R/O	EnumRange R/O (OFF ON)
state.fan.2	CHOICE	R/O	EnumRange R/O (OFF ON)
state.cool.1	CHOICE	R/O	EnumRange R/O (OFF ON)
state.cool.2	CHOICE	R/O	EnumRange R/O (OFF ON)
state.cool.3	CHOICE	R/O	EnumRange R/O (OFF ON)
state.cool.4	CHOICE	R/O	EnumRange R/O (OFF ON)
state.heat.1	CHOICE	R/O	EnumRange R/O (OFF ON)
state.heat.2	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
state.heat.3	CHOICE	R/O	EnumRange R/O (OFF ON)
state.heat.4	CHOICE	R/O	EnumRange R/O (OFF ON)
state.dehumidifying	CHOICE	R/O	EnumRange R/O (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fire Smoke Shitdown)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Controller Failure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Second Stage Cooling)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High Temperature 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (High Temperature 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power Loss Unit 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power Loss Unit 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Unit Lockout 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit Lockout 2)

9.79 Blankom-DRP393-SNMP

This page describes the device driver and the device window for the Blankom DRP-393 IRD, controlled via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains MPTS transport stream output parameters (ASI and IP)
- --- Contains the parameters concerning the video decoding and SDI output.

- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DRP-393.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbVoltage --- Selects the LNB power supply voltage (OFF/13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

- bandEdge --- The frequency where the software switches the LO frequencies.
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.productName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.bootloaderVersion	TEXT	R/O	StringRange R/O
info.applicationVersion	TEXT	R/O	StringRange R/O
info.decoderVersion	TEXT	R/O	StringRange R/O
info.fpgaVersion	TEXT	R/O	StringRange R/O
info.sdiFpgaVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input	CHOICE		EnumRange (ASI SAT IP)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (1.0000 .. 45.0000)
dvbmode	CHOICE		EnumRange (AUTO DVB-S DVB-S2)
polarization	CHOICE		EnumRange (X Y)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
programNo	INTEGER		IntegerRange (0 .. 256)
programList	CHOICE		EnumRange (0 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
tsout.ipOut	CHOICE		EnumRange (OFF ON)
tsout.ipProtocol	CHOICE		EnumRange (UDP RTP PRO-MPEG)
tsout.ipAddress	TEXT		StringRange

name	*type*	*flags*	*range*
tsout.ipPort	INTEGER		IntegerRange (1 .. 65535)
tsout.asiOut	CHOICE		EnumRange (OFF ON)
tsout.dataRateMode	CHOICE		EnumRange (INPUT CBR-USER-DEFINED VBR-NULL-PACKET-DELETION)
tsout.dataRate	FLOAT		DoubleRange (0.000 .. 0.000)
video.resolution	TEXT	R/O	StringRange R/O
video.aspect	TEXT	R/O	StringRange R/O
video.encoding	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
video.colorBars	CHOICE		EnumRange (OFF ON)
video.arDownConvert	CHOICE		EnumRange (4:3-LETTERBOX 16:9-BYPASS)
video.format	CHOICE		EnumRange (PAL SECAM NTSC)
video.framerate	TEXT	R/O	StringRange R/O
video.sdi.mode	CHOICE		EnumRange (SD SD/HD)
video.sdi.audio1	CHOICE		EnumRange (OFF ON)
video.sdi.audio2	CHOICE		EnumRange (OFF ON)
video.sdi.aesAudio	CHOICE		EnumRange (AUDIO-1 AUDIO-2)
audio.testTone	CHOICE		EnumRange (OFF ON)
audio.testOutput	CHOICE		EnumRange (OUTPUT-1 OUTPUT-2 OUTPUT-3 OUTPUT-4)
audio.1.dualChannel	CHOICE		EnumRange (OFF PERMANENT EN301775-DC-STATUS-ONLY)

name	*type*	*flags*	*range*
audio.1.gain	CHOICE		EnumRange (MUTE +10.0 dB +9.5 dB +9.0 dB +8.5 dB +8.0 dB +7.5 dB +7.0 dB +6.5 dB +6.0 dB +5.5 dB +5.0 dB +4.5 dB +4.0 dB +3.5 dB +3.0 dB +2.5 dB +2.0 dB +1.5 dB +1.0 dB +0.5 dB +0.0 dB -0.5 dB -1.0 dB -1.5 dB -2.0 dB -2.5 dB -3.0 dB -3.5 dB -4.0 dB -4.5 dB -5.0 dB -5.5 dB -6.0 dB -6.5 dB -7.0 dB -7.5 dB -8.0 dB -8.5 dB -9.0 dB -9.5 dB -10.0 dB -10.5 dB -11.0 dB -11.5 dB -12.0 dB -12.5 dB -13.0 dB -13.5 dB -14.0 dB -14.5 dB -15.0 dB -15.5 dB -16.0 dB -16.5 dB -17.0 dB -17.5 dB -18.0 dB -18.5 dB -19.0 dB -19.5 dB -20.0 dB -20.5 dB -21.0 dB -21.5 dB -22.0 dB -22.5 dB -23.0 dB -23.5 dB -24.0 dB -24.5 dB -25.0 dB -25.5 dB -26.0 dB -26.5 dB -27.0 dB -27.5 dB -28.0 dB -28.5 dB -29.0 dB -29.5 dB -30.0 dB)
audio.2.gain	CHOICE		EnumRange (MUTE +10.0 dB +9.5 dB +9.0 dB +8.5 dB +8.0 dB +7.5 dB +7.0 dB +6.5 dB +6.0 dB +5.5 dB +5.0 dB +4.5 dB +4.0 dB +3.5 dB +3.0 dB +2.5 dB +2.0 dB +1.5 dB +1.0 dB +0.5 dB +0.0 dB -0.5 dB -1.0 dB -1.5 dB -2.0 dB -2.5 dB -3.0 dB -3.5 dB -4.0 dB -4.5 dB -5.0 dB -5.5 dB -6.0 dB -6.5 dB -7.0 dB -7.5 dB -8.0 dB -8.5 dB -9.0 dB -9.5 dB -10.0 dB -10.5 dB -11.0 dB -11.5 dB -12.0 dB -12.5 dB -13.0 dB -13.5 dB -14.0 dB -14.5 dB -15.0 dB -15.5 dB -16.0 dB -16.5 dB -17.0 dB -17.5 dB -18.0 dB -18.5 dB -19.0 dB -19.5 dB -20.0 dB -20.5 dB -21.0 dB -21.5 dB -22.0 dB -22.5 dB -23.0 dB -23.5 dB -24.0 dB -24.5 dB -25.0 dB -25.5 dB -26.0 dB -26.5 dB -27.0 dB -27.5 dB -28.0 dB -28.5 dB -29.0 dB -29.5 dB -30.0 dB)

name	*type*	*flags*	*range*
audio.3.gain	CHOICE		EnumRange (MUTE +10.0 dB +9.5 dB +9.0 dB +8.5 dB +8.0 dB +7.5 dB +7.0 dB +6.5 dB +6.0 dB +5.5 dB +5.0 dB +4.5 dB +4.0 dB +3.5 dB +3.0 dB +2.5 dB +2.0 dB +1.5 dB +1.0 dB +0.5 dB +0.0 dB -0.5 dB -1.0 dB -1.5 dB -2.0 dB -2.5 dB -3.0 dB -3.5 dB -4.0 dB -4.5 dB -5.0 dB -5.5 dB -6.0 dB -6.5 dB -7.0 dB -7.5 dB -8.0 dB -8.5 dB -9.0 dB -9.5 dB -10.0 dB -10.5 dB -11.0 dB -11.5 dB -12.0 dB -12.5 dB -13.0 dB -13.5 dB -14.0 dB -14.5 dB -15.0 dB -15.5 dB -16.0 dB -16.5 dB -17.0 dB -17.5 dB -18.0 dB -18.5 dB -19.0 dB -19.5 dB -20.0 dB -20.5 dB -21.0 dB -21.5 dB -22.0 dB -22.5 dB -23.0 dB -23.5 dB -24.0 dB -24.5 dB -25.0 dB -25.5 dB -26.0 dB -26.5 dB -27.0 dB -27.5 dB -28.0 dB -28.5 dB -29.0 dB -29.5 dB -30.0 dB)
audio.4.gain	CHOICE		EnumRange (MUTE +10.0 dB +9.5 dB +9.0 dB +8.5 dB +8.0 dB +7.5 dB +7.0 dB +6.5 dB +6.0 dB +5.5 dB +5.0 dB +4.5 dB +4.0 dB +3.5 dB +3.0 dB +2.5 dB +2.0 dB +1.5 dB +1.0 dB +0.5 dB +0.0 dB -0.5 dB -1.0 dB -1.5 dB -2.0 dB -2.5 dB -3.0 dB -3.5 dB -4.0 dB -4.5 dB -5.0 dB -5.5 dB -6.0 dB -6.5 dB -7.0 dB -7.5 dB -8.0 dB -8.5 dB -9.0 dB -9.5 dB -10.0 dB -10.5 dB -11.0 dB -11.5 dB -12.0 dB -12.5 dB -13.0 dB -13.5 dB -14.0 dB -14.5 dB -15.0 dB -15.5 dB -16.0 dB -16.5 dB -17.0 dB -17.5 dB -18.0 dB -18.5 dB -19.0 dB -19.5 dB -20.0 dB -20.5 dB -21.0 dB -21.5 dB -22.0 dB -22.5 dB -23.0 dB -23.5 dB -24.0 dB -24.5 dB -25.0 dB -25.5 dB -26.0 dB -26.5 dB -27.0 dB -27.5 dB -28.0 dB -28.5 dB -29.0 dB -29.5 dB -30.0 dB)

name	*type*	*flags*	*range*
audio.spdif	CHOICE		EnumRange (DISABLED BYPASS DECODED-DOWNMIX)
audio.downmix	CHOICE		EnumRange (DOLBY-SURROUND-PROLOGIC STEREO)
audio.drCompression	CHOICE		EnumRange (LINE-MODE RF-MODE EXPERT-MODE)
audio.1.stream	CHOICE		EnumRange (0 - NO AUDIO)
audio.2.stream	CHOICE		EnumRange (0 - NO AUDIO)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
state.dvbmode	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.band	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.80 Brainbox-EDxxx

This page describes the device driver and the device window for the*Brainbox ED series* data acquisition module family. The device driver monitors and controls the digital inputs/outputs of various Brainbox EDxx modules. It supports:

- ED004 - 4 digital inputs/outputs
- ED008 - 8 digital inputs/outputs

- ED024 - 4 digital inputs/outputs
- ED588 - 8 digital inputs/outputs
- ED516 - 16 digital inputs

It was tested with a ED-508

- SW Version: 6.15

The driver does not support: ED-527,ED-538,ED-038 modules.

The *Brainbox-EDxxx* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the *Brainbox-EDxxx* device driver directly.

There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Brainbox-EDxxx* device driver as a low level driver to access the physical I/O ports.

Input ports

The variable inputs.0 refers to the input ports provided by the device. The least significant bit denotes the state of DI0.

The variable reports the physical state if the input. A bit is set to '1' if the input is open or externally driven to high level. The bit reads '0' if the input is connected to GND or externally driven to low level.

Output ports

The variable 'outputs.0' reports the actual state of the digital outputs and also sets the output states. The least significant bit in the variable corresponds to output DO0.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Brainbox ED module acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the device in the device setup window , there must not be defined any other devices beside this module for this interface.
- The Brainbox ED modules are controlled through Terminal-CR-CR protocol. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:9500" where 9500 is the standard TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

The unit has to be configured to use ASCII protocol with the following settings:

- Address 1 (Hex 01)
- Checksum = Disabled

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1
moduleType	The actual module type must be selected to tell the driver how many inputs/outputs are available with this module.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
config.moduleType	CHOICE	SAVE SETUP	EnumRange (ED004 ED008 ED204 ED508 ED516)
inputs.0	HEX	R/O	HexRange R/O (0 .. 0)
outputs.0	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange

9.81 CGC-ACU

This page describes the device driver and the device window for the CGC ACU via HTTP11.

The default TCP/IP communication port of the ACU is 19191.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the measurements and tracking state.
- --- Contains the deice parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	Add the address as well as the remote access port. Default port is 19191 e.g. 10.10.1.10:19191
ftp.user	username for ftp authentification used for ftp TLE/OEM upload
ftp.password	password for ftp authentification used for ftp TLE/OEM upload

parameter	*description*
ftp.ipaddress	ip address used for ftp TLE/OEM upload
hasFeedblower	Select if unit is equipped with a feedblower
hasDeice	Select if unit is equipped with a deice system
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.session	TEXT	R/O	StringRange R/O
info.postResponse	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.upperCastingTemp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.lowerCastingTemp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.upperDriveTemp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.lowerDriverTemp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.upperDriveVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.lowerDriveVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.upperDriveCurrent	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.lowerDriveCurrent	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.freqLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.freqHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.s	TEXT	R/O	StringRange R/O
internal.p	TEXT	R/O	StringRange R/O
internal.a	TEXT	R/O	StringRange R/O
internal.tStart	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tEnd	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.version	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.version2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.satellites	TEXT	R/O	StringRange R/O
internal.autoSatellites	TEXT	R/O	StringRange R/O
config.ftp.user	TEXT	SAVE SETUP	StringRange
config.ftp.password	TEXT	SAVE SETUP	StringRange
config.tleUploadDir	TEXT	SAVE SETUP	StringRange
config.oemUploadDir	TEXT	SAVE SETUP	StringRange
config.hasFeedblower	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.hasDeice	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.cpu.highLoadThd	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.mem.highLoadThd	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.disk.highLoadThd	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
cpu.load	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
mem.load	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
disk.load	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.attn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.bandwidth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.averaging	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dtr.frequency	FLOAT		DoubleRange (945.000 .. 20300.000)
dtr.attn	INTEGER		IntegerRange (0 .. 30)

name	*type*	*flags*	*range*
dtr.bandwidth	INTEGER		IntegerRange (6 .. 100)
dtr.averaging	FLOAT		DoubleRange (1.0 .. 50.0)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.brake	BOOLEAN	R/O	BooleanRange R/O (true false)
state.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
state.progress	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.model	BOOLEAN	R/O	BooleanRange R/O (true false)
state.azOffset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.elOffset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.timeOffset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.upperXVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.lowerYVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.mode4	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	CHOICE	R/O	EnumRange R/O (RHCP LHCP V H)
updateDest	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange
zenith	TEXT	NOPRESET	StringRange
clear	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
tracking	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
profile	CHOICE		EnumRange (none)
uploadTLE	TEXT	NOPRESET	StringRange
uploadOEM	TEXT	NOPRESET	StringRange
uploadSuccess	BOOLEAN	R/O	BooleanRange R/O (true false)
state.offset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.deviation	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
events	TEXT	NOPRESET NOCOMPARE	StringRange
schedule.manual	TEXT	NOPRESET	StringRange
schedule.auto	TEXT	NOPRESET	StringRange
schedule.enable	TEXT	NOPRESET	StringRange
schedule.disable	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
schedule.add	TEXT	NOPRESET	StringRange
auto.track	CHOICE	NOPRESET	EnumRange ()
refresh.ephemeris.files	TEXT	NOPRESET	StringRange
feedblower.state	BOOLEAN	R/O	BooleanRange R/O (true false)
humidity	CHOICE	R/O	EnumRange R/O (DRY WET)
flags.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.main.thermostat	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.main.heater	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.main.feedback	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.main.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
deice.main.test	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.main.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
deice.main.temp.on	FLOAT		DoubleRange (0.0 .. 0.0)
deice.main.temp.off	FLOAT		DoubleRange (0.0 .. 0.0)
deice.main.mode	CHOICE		EnumRange (OFF AUTO)
deice.main.mode2	CHOICE		EnumRange (OFF ON)
deice.main.test.stop	TEXT	NOPRESET	StringRange
deice.main.test.5m	TEXT	NOPRESET	StringRange
deice.main.test.10m	TEXT	NOPRESET	StringRange
deice.sub.thermostat	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.sub.heater	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
deice.sub.feedback	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.sub.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
deice.sub.test	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.sub.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
deice.sub.temp.on	FLOAT		DoubleRange (0.0 .. 0.0)
deice.sub.temp.off	FLOAT		DoubleRange (0.0 .. 0.0)
deice.sub.mode	CHOICE		EnumRange (OFF AUTO)
deice.sub.mode2	CHOICE		EnumRange (OFF ON)
deice.sub.test.stop	TEXT	NOPRESET	StringRange
deice.sub.test.5m	TEXT	NOPRESET	StringRange
deice.sub.test.10m	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Upper/X east axis)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Lower/Y north axis)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Upper axis)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Lower axis)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Upper axis board)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Lower axis board)
faults.08	ALARM	R/O	AlarmFlagRange R/O (DTR)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Test flag set)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (RemoteEmergency stop)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Upper axis drive)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Lower axis drive)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Upper axis not ready)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Lower axis not ready)
faults.17	ALARM	R/O	AlarmFlagRange R/O (NTP time synch lost)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Upper casting < -40C)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Upper casting > 60C)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Lower casting < -40C)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Lower casting > 60C)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Upper drive < -40C)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Upper drive > 60C)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Lower drive < -40C)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Lower drive > 60C)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Up.axis soft lim.disable)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Upper axis hard limit)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (Upper axis final limit)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Lo.axis soft lim.disable)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Lo.axis hard limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Lo.axis final limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Up.hard limit override)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Lo.hard limit override)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Final limit override)
faults.35	ALARM	R/O	AlarmFlagRange R/O (High Disk load)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Beacon Lock)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Inclinometer)
faults.38	ALARM	R/O	AlarmFlagRange R/O (PMCU)
faults.39	ALARM	R/O	AlarmFlagRange R/O (PMCU emergency stop)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Rain blower pwr signal)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Auto-track lock)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Upper cooling module)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Upper hand crank)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Lower cooling module)

name	*type*	*flags*	*range*
faults.45	ALARM	R/O	AlarmFlagRange R/O (Lower hand crank)
faults.46	ALARM	R/O	AlarmFlagRange R/O (PMCU was connect.)
faults.47	ALARM	R/O	AlarmFlagRange R/O (PMCU was disconnect.)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Deice emergency stop)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Main deice RTD probe)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Sub deice RTD probe)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Deice RTD)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Deice DIO)
faults.54	ALARM	R/O	AlarmFlagRange R/O (CPU Load)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Memory Load)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Disk Load)

9.82 CPI-350W-TWTA

This page describes the device driver and the device window for the CPI 350W CMPA amplifier. This type does not work with the standard CPI-TWTA device driver. Unlike the recent amplifier models from CPI, this type expects the attenuation as a (PIN) voltage parameter rather than as a attenuation expressed in dB.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
pinVoltage	The voltage to applied at the pin diode attenuator which is built into the CMPA
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.checkLimits	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.pinVoltage	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 20.00)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.helixAdvWarning	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power fail timeout)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC buss under voltage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Helix voltage in standby)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.21	ALARM	R/O	AlarmFlagRange R/O (SSIPA over current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.83 CPI-CHPA

This page describes the device driver and the device window for the CPI compact high power amplifiers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal).
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Configuration notes

- select RS232 or RS442/485 4wire via slide switch mounted inside the unit
- connector for M&C is J7 CIF Computer Interface
- protocol version : CIF configuration => protocol => STX/ETX.
- address byte : CIF configuration => port address => xx
- check byte : CIF configuration => checkbyte => longitudinal parity
- cr/lf : CIF configuration => CR enabled / LF enabled => no
- DIP Switch S1 Switch #2. => OFF (default) : attenuator setting is in the form NN.N. => ON : attenuator setting is in the form NN.N0 (compatible to LHPA)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.checkLimits	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.hasElapsed	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.unitMeasLevel	CHOICE	SAVE SETUP	EnumRange (dBm dBW)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 25.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 25.0)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (0.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iBeam	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (BB RAM)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF peak output power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power fail timeout)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC buss under voltage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Helix voltage in standby)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (External interlock)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (SSIPA over current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.84 CPI-GaN-SSPA

This page describes the device driver and the device window for the CPI GaN SSPA using binary BCIP protocol. It is prepared to support up to 16 WG switches. Redundancy and WG switches are not tested.

The driver was tested with the following models:

- B5KO (160W Ka Band)
- FW: 01.03.24 (Main)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters.
- --- Displays meter readings like output power and temperatures.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	*description*
address	The device address (decimal). The factory default is 48
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
loRfOutAlarm	The low output power alarm trip point.
hiRfOutAlarm	The high output power alarm trip point.
loRfOutFault	The low output power fault trip point.
hiRfOutFault	The high output power fault trip point.
hiRefOutAlarm	The high reflected power alarm trip point.
hiRefOutFault	The high reflected power fault trip point.
redundancy	Defines the redundancy mode.
switches	Defines the number of switches.
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
gainRange	Defines the available gain range of the unit (Default 30dB).
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" in the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.amptype	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.muc.main.rev	TEXT	R/O	StringRange R/O
info.muc.co.kernel.rev	TEXT	R/O	StringRange R/O
info.muc.co.main.rev	TEXT	R/O	StringRange R/O
info.rfDet.kernel.rev	TEXT	R/O	StringRange R/O
info.rfDet.main.rev	TEXT	R/O	StringRange R/O
info.rfDri.kernel.rev	TEXT	R/O	StringRange R/O
info.rfDri.main.rev	TEXT	R/O	StringRange R/O
info.led.kernel.rev	TEXT	R/O	StringRange R/O
info.led.main.rev	TEXT	R/O	StringRange R/O
info.dbs.kernel.rev	TEXT	R/O	StringRange R/O
info.dbs.main.rev	TEXT	R/O	StringRange R/O
info.uPA.kernel.rev	TEXT	R/O	StringRange R/O
info.uPA.main.rev	TEXT	R/O	StringRange R/O
info.buc.kernel.rev	TEXT	R/O	StringRange R/O
info.buc.main.rev	TEXT	R/O	StringRange R/O
info.muc.boot.kernel.rev	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.hiRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.loRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.hiRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.hiRefOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.hiRefOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1)
config.switches	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.alc	CHOICE		EnumRange (MANUAL ALC)
tx.requestedLevel	FLOAT		DoubleRange (-100.0 .. 40.0)
state	TEXT	R/O	StringRange R/O
state2	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.uDriver	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iDriver	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uBooster	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iBooster	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.iPsu	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.hotspotTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cabTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.driverTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.boosterTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.dbsTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fan.a1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.a2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.a3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.a4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.b1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.b2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.b3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.b4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iFan	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fan.bank.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fan.bank.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hours.unit	TEXT	R/O	StringRange R/O
hours.bias	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange (1 2)
position.01	CHOICE		EnumRange DIS (A B)

name	*type*	*flags*	*range*
position.02	CHOICE		EnumRange DIS (A B)
position.03	CHOICE		EnumRange DIS (A B)
position.04	CHOICE		EnumRange DIS (A B)
position.05	CHOICE		EnumRange DIS (A B)
position.06	CHOICE		EnumRange DIS (A B)
position.07	CHOICE		EnumRange DIS (A B)
position.08	CHOICE		EnumRange DIS (A B)
position.09	CHOICE		EnumRange DIS (A B)
position.10	CHOICE		EnumRange DIS (A B)
position.11	CHOICE		EnumRange DIS (A B)
position.12	CHOICE		EnumRange DIS (A B)
position.13	CHOICE		EnumRange DIS (A B)
position.14	CHOICE		EnumRange DIS (A B)
position.15	CHOICE		EnumRange DIS (A B)
position.16	CHOICE		EnumRange DIS (A B)
internal.f02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f11	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.f12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.days	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (BBRAM)
faults.02	ALARM	R/O	AlarmFlagRange R/O (CAN Bus)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CAN Level Key)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (M&C Co Controller)
faults.05	ALARM	R/O	AlarmFlagRange R/O (BUC Unlock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (System)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Sem Comm)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver Comm)
faults.09	ALARM	R/O	AlarmFlagRange R/O (RF Det Comm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (BUC Comm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LED Panel Comm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DBS Comm)
faults.13	ALARM	R/O	AlarmFlagRange R/O (UPA 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (UPA 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (UPA 3)
faults.16	ALARM	R/O	AlarmFlagRange R/O (UPA 4)
faults.17	ALARM	R/O	AlarmFlagRange R/O (UPA 5)
faults.18	ALARM	R/O	AlarmFlagRange R/O (UPA 6)
faults.19	ALARM	R/O	AlarmFlagRange R/O (UPA 7)
faults.20	ALARM	R/O	AlarmFlagRange R/O (UPA 8)
faults.21	ALARM	R/O	AlarmFlagRange R/O (UPA 9)
faults.22	ALARM	R/O	AlarmFlagRange R/O (UPA 10)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Fan Bank A)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Fan Bank B)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Cabinet under temp)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temp)

name	*type*	*flags*	*range*
faults.27	ALARM	R/O	AlarmFlagRange R/O (M&C CO EEPROM)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Ext Interlock)
faults.29	ALARM	R/O	AlarmFlagRange R/O (M&C CO System Comm)
faults.30	ALARM	R/O	AlarmFlagRange R/O (RF Det EEPROM)
faults.31	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.33	ALARM	R/O	AlarmFlagRange R/O (High Reflected)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Hotspot under temp)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Hotspot over temp)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver EEPROM)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver I2C Bus)
faults.38	ALARM	R/O	AlarmFlagRange R/O (LED Panel over temp)
faults.39	ALARM	R/O	AlarmFlagRange R/O (LED Panel Comm)
faults.40	ALARM	R/O	AlarmFlagRange R/O (DBS under temp)
faults.41	ALARM	R/O	AlarmFlagRange R/O (DBS over temp)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Fan A under volt)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Fan A over volt)

name	*type*	*flags*	*range*
faults.44	ALARM	R/O	AlarmFlagRange R/O (Fan A over current)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Fan B under volt)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Fan B over volt)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Fan B over current)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 5V under volt)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 5V over volt)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 5V under curr)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 5V over curr)
faults.52	ALARM	R/O	AlarmFlagRange R/O (House Keep 12V under volt)
faults.53	ALARM	R/O	AlarmFlagRange R/O (House Keep 12V over volt)
faults.54	ALARM	R/O	AlarmFlagRange R/O (House Keep 12V over curr)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Driver)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Booster)
faults.57	ALARM	R/O	AlarmFlagRange R/O (M&C C Board over temp)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Low Battery)
faults.59	ALARM	R/O	AlarmFlagRange R/O (RTC Failure)
faults.60	ALARM	R/O	AlarmFlagRange R/O (BUC Lock)
faults.61	ALARM	R/O	AlarmFlagRange R/O (3.3V under volt)

name	*type*	*flags*	*range*
faults.62	ALARM	R/O	AlarmFlagRange R/O (5V under volt)
faults.63	ALARM	R/O	AlarmFlagRange R/O (12V under volt)
faults.64	ALARM	R/O	AlarmFlagRange R/O (24V under volt)
faults.65	ALARM	R/O	AlarmFlagRange R/O (SD Flash)
faults.66	ALARM	R/O	AlarmFlagRange R/O (BUC VCC under volt)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Lin VCC under volt)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Reference Toggle)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Temp Compensation data)
faults.70	ALARM	R/O	AlarmFlagRange R/O (PFM1)
faults.71	ALARM	R/O	AlarmFlagRange R/O (PFM2)
faults.72	ALARM	R/O	AlarmFlagRange R/O (PFM3)
faults.73	ALARM	R/O	AlarmFlagRange R/O (PFM4)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Linearizer)
faults.75	ALARM	R/O	AlarmFlagRange R/O (External Inhibit)
faults.76	ALARM	R/O	AlarmFlagRange R/O (BUC PLL Lock)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Fan A1)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Fan A2)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Fan A3)
faults.80	ALARM	R/O	AlarmFlagRange R/O (Fan A4)
faults.81	ALARM	R/O	AlarmFlagRange R/O (Fan B1)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Fan B2)

name	*type*	*flags*	*range*
faults.83	ALARM	R/O	AlarmFlagRange R/O (Fan B3)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Fan B4)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Fan Under Speed)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Fan Over Speed)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Fan Stall)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Fan Idling)
faults.89	ALARM	R/O	AlarmFlagRange R/O (ALC Limit)
faults.90	ALARM	R/O	AlarmFlagRange R/O (FWD Power Detector)
faults.91	ALARM	R/O	AlarmFlagRange R/O (REV Power Detector)
faults.92	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver Under volt)
faults.93	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 1 under curr)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 2 under curr)
faults.95	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver 3 under curr)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver Vgs under volt)
faults.97	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver VCC under volt)
faults.98	ALARM	R/O	AlarmFlagRange R/O (Pre-Driver VD2 under volt)
faults.100	ALARM	R/O	AlarmFlagRange R/O (Switch Inhibit)
faults.101	ALARM	R/O	AlarmFlagRange R/O (Ethernet Inhibit)

name	*type*	*flags*	*range*
faults.102	ALARM	R/O	AlarmFlagRange R/O (Serial (CIF) Inhibit)
faults.103	ALARM	R/O	AlarmFlagRange R/O (Local Inhibit)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Bootloader incompatible)
faults.105	ALARM	R/O	AlarmFlagRange R/O (AC Power Fail Fault)
faults.106	ALARM	R/O	AlarmFlagRange R/O (Ethernet CIF inhibit)
faults.107	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.108	ALARM	R/O	AlarmFlagRange R/O (Switch 1 comm)
faults.109	ALARM	R/O	AlarmFlagRange R/O (Switch 1 stall)
faults.110	ALARM	R/O	AlarmFlagRange R/O (Switch 2 comm)
faults.111	ALARM	R/O	AlarmFlagRange R/O (Switch 2 stall)
faults.112	ALARM	R/O	AlarmFlagRange R/O (Switch 3 comm)
faults.113	ALARM	R/O	AlarmFlagRange R/O (Switch 3 stall)
faults.114	ALARM	R/O	AlarmFlagRange R/O (Switch 4 comm)
faults.115	ALARM	R/O	AlarmFlagRange R/O (Switch 4 stall)
faults.116	ALARM	R/O	AlarmFlagRange R/O (Switch 5 comm)
faults.117	ALARM	R/O	AlarmFlagRange R/O (Switch 5 stall)
faults.118	ALARM	R/O	AlarmFlagRange R/O (Switch 6 comm)

name	*type*	*flags*	*range*
faults.119	ALARM	R/O	AlarmFlagRange R/O (Switch 6 stall)
faults.120	ALARM	R/O	AlarmFlagRange R/O (Switch 7 comm)
faults.121	ALARM	R/O	AlarmFlagRange R/O (Switch 7 stall)
faults.122	ALARM	R/O	AlarmFlagRange R/O (Switch 8 comm)
faults.123	ALARM	R/O	AlarmFlagRange R/O (Switch 8 stall)
faults.124	ALARM	R/O	AlarmFlagRange R/O (Switch 9 comm)
faults.125	ALARM	R/O	AlarmFlagRange R/O (Switch 9 stall)
faults.126	ALARM	R/O	AlarmFlagRange R/O (Switch 10 comm)
faults.127	ALARM	R/O	AlarmFlagRange R/O (Switch 10 stall)
faults.128	ALARM	R/O	AlarmFlagRange R/O (Switch 11 comm)
faults.129	ALARM	R/O	AlarmFlagRange R/O (Switch 11 stall)
faults.130	ALARM	R/O	AlarmFlagRange R/O (Switch 12 comm)
faults.131	ALARM	R/O	AlarmFlagRange R/O (Switch 12 stall)
faults.132	ALARM	R/O	AlarmFlagRange R/O (Switch 13 comm)
faults.133	ALARM	R/O	AlarmFlagRange R/O (Switch 13 stall)
faults.134	ALARM	R/O	AlarmFlagRange R/O (Switch 14 comm)

name	*type*	*flags*	*range*
faults.135	ALARM	R/O	AlarmFlagRange R/O (Switch 14 stall)
faults.136	ALARM	R/O	AlarmFlagRange R/O (Switch 15 comm)
faults.137	ALARM	R/O	AlarmFlagRange R/O (Switch 15 stall)
faults.138	ALARM	R/O	AlarmFlagRange R/O (Switch 16 comm)
faults.139	ALARM	R/O	AlarmFlagRange R/O (Switch 16 stall)

9.85 CPI-KHPA-G3

This page describes the device driver and the device window for the CPI Klystron amplifiers Generation 3.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and Klystron specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*
address	The device address (decimal). The factory default is 48
lowLimitAlarm	The low output power alarm trip point.
highLimitAlarm	The high output power alarm trip point.
lowLimitFault	The low output power fault trip point.

parameter	*description*
highLimitFault	The high output power fault trip point.
Serial	Remember to set the line parameters for the serial interface. of the HPA is *9600,E,7,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control

Serial Interface

The standard configuration is RS-422. To configure RS-232 use the jumpers described in the user manual of the Klystron Chapter 3.

You can configure the line setting on the front panel.

Channel Changer

If the Klystron is equipped with an channel changer you have to configure the number of channels on the front panel of the HPA. Then the M&C system is able to detect this via the ID/Version Query and the driver will enable the channel field. If the Klystron is not equipped with an channel changer, set the number of channels on the front panel to 0 and the driver will not allow to select a channel.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.maxChannels	INTEGER	R/O	IntegerRange R/O (0 .. 24)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimitAlarm	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.highLimitAlarm	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.lowLimitFault	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.highLimitFault	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.channelSelect	CHOICE		EnumRange DIS (0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 25.5)
tx.attn	FLOAT		DoubleRange (0.0 .. 25.5)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 40.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
tx.adjustInProgress	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.uBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 100.00)
meas.iBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 10.00)
meas.iBody	FLOAT	R/O	DoubleRange R/O (0.0 .. 1000.0)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)

name	*type*	*flags*	*range*
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.drive	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.maxChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Heater not ready)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.05	ALARM	R/O	AlarmFlagRange R/O (Heater time delay)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Beam overcurrent)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Body overcurrent)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Heater pwr.supply)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Beam regulator)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Safety interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low Air Flow)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External (latched))
faults.13	ALARM	R/O	AlarmFlagRange R/O (Reflected power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Waveguide)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Circulator Overload)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Collector Overtemp)
faults.17	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RF-Switch inhibit)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Beam regulator bypass)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.86 CPI-KHPA-G4

This page describes the device driver and the device window for the CPI Klystron amplifiers Generation 4.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)

- --- Displays meter readings like output power and Klystron specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
lowLimitAlarm	The low output power alarm trip point.
highLimitAlarm	The high output power alarm trip point.
lowLimitFault	The low output power fault trip point.
highLimitFault	The high output power fault trip point.
Serial	Remember to set the line parameters for the serial interface. of the HPA is *9600,E,7,1* .
redundancy	Configures the operation mode of the internal redundancy switching logic of the amplifier.
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control

Serial Interface

You can configure the line setting on the front panel. This includes also the selection of RS232 or RS485.

Channel Changer

If the Klystron is equipped with an channel changer you have to configure the number of channels on the front panel of the HPA. Then the M&C system is able to detect this via the ID/Version Query and the driver will enable the channel field. If the Klystron is not equipped with an channel changer, set the number of channels on the front panel to 0 and the driver will not allow to select a channel.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.maxChannels	INTEGER	R/O	IntegerRange R/O (0 .. 50)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimitAlarm	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.highLimitAlarm	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.lowLimitFault	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.highLimitFault	INTEGER	SETUP	IntegerRange (0 .. 3100)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1 2:1 MAIN 2:1 BACKUP)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.channelSelect	CHOICE		EnumRange DIS (0)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.beam	CHOICE		EnumRange (STANDBY 1KV STBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 25.5)
tx.attn	FLOAT		DoubleRange (0.0 .. 25.5)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 40.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
tx.adjustInProgress	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.uBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 100.00)
meas.iBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 10.00)
meas.iBody	FLOAT	R/O	DoubleRange R/O (0.0 .. 1000.0)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.drive	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.inletTemp	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.outletTemp	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
protection	CHOICE		EnumRange (MANUAL AUTO)
position.01	CHOICE		EnumRange (A B N/A)
position.02	CHOICE		EnumRange (A B N/A)
position.03	CHOICE		EnumRange (A B N/A)
position.04	CHOICE		EnumRange (A B N/A)
position.05	CHOICE		EnumRange (A B N/A)
position.06	CHOICE		EnumRange (A B N/A)

name	*type*	*flags*	*range*
position.07	CHOICE		EnumRange (A B N/A)
position.08	CHOICE		EnumRange (A B N/A)
position.09	CHOICE		EnumRange (A B N/A)
position.10	CHOICE		EnumRange (A B N/A)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.maxChannels	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Beam over voltage)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Beam under voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Beam over current)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Beam under current)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Beam during stdby)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Body over current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Heater over voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Heater under voltage)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Heater under current)
faults.13	ALARM	R/O	AlarmFlagRange R/O (High voltage ARC)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High voltage Interlock)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Lost AC)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Lost AC phase)
faults.17	ALARM	R/O	AlarmFlagRange R/O (15V under voltage)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PS Blower)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PS heatsink temp)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Circulator over temp)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Collector over temp)
faults.22	ALARM	R/O	AlarmFlagRange R/O (High reflected RF)
faults.23	ALARM	R/O	AlarmFlagRange R/O (High RF fault)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Low RF fault)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Klystron WG Arc 1)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Channel changer)
faults.27	ALARM	R/O	AlarmFlagRange R/O (High outlet temp)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Low airflow)
faults.29	ALARM	R/O	AlarmFlagRange R/O (High inlet humitidy)
faults.30	ALARM	R/O	AlarmFlagRange R/O (High outlet temp)
faults.31	ALARM	R/O	AlarmFlagRange R/O (High RF alarm)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Low RF alarm)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RF-Switch inhibit)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.87 CPI-LPA

This page describes the device driver and the device window for the CPI Low Power TWT amplifiers. This driver was tested with 125W LPA HUB MOUNT AMPLIFIER Typ VZU6992v3

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 900000)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 900000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 25.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 25.0)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (BB RAM)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power fail timeout)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC buss under voltage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Helix voltage in standby)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.21	ALARM	R/O	AlarmFlagRange R/O (SSIPA over current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.88 CPI-Mini-ODU

This page describes the device driver and the device window for the CPI Mini ODU power amplifier (models T5KO/TL05KO).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The redundancy parameters.
- --- The uplink power control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
lowLimitAlarm	The low output power alarm trip point.
highLimitAlarm	The high output power alarm trip point.
lowLimitFault	The low output power fault trip point.
highLimitFault	The high output power fault trip point.
hiRefOutAlarm	The reflected power alarm trip point.
hiRefOutFault	The reflected power fault trip point.
hiRfInpAlarm	The high input power alarm trip point.
hiRfInpFault	The high input power fault trip point.
redundancy	Defines the redundancy type.
switches	Defines the number of installed switches 0-6.
upcExtInhibit	Enables or disables the beacon receiver fail alarm indication.

parameter	*description*
nominalPosSw1	Defines the nominal position of switch 1 in a 2:1 redundancy.
nominalPosSw2	Defines the nominal position of switch 2 in a 2:1 redundancy.
nominalPosSw3	Defines the nominal position of switch 3 in a 2:1 redundancy.
nominalPosSw4	Defines the nominal position of switch 4 in a 2:1 redundancy.
Serial	Remember to set the line parameters for the serial interface. of the HPA is *9600,E,7,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" int the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.amptype	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.main.kernel.rev	TEXT	R/O	StringRange R/O
info.main.panel.rev	TEXT	R/O	StringRange R/O
info.controlPoint	TEXT	R/O	StringRange R/O
info.switchSystem	TEXT	R/O	StringRange R/O DIS
info.swSystemAmpId	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
info.hasUPC	CHOICE	R/O	EnumRange R/O (NO YES)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.loRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRefOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRefOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfInpAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfInpFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1 2:1)
config.switches	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6)
config.upcExtInhibit	CHOICE	SETUP	EnumRange (ENABLED DISABLED)

name	*type*	*flags*	*range*
config.nominalPosSw1	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw2	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw3	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw4	CHOICE	SAVE SETUP	EnumRange (A B)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.alc	CHOICE		EnumRange (MANUAL ALC UPC)
tx.requestedLevel	FLOAT		DoubleRange (-100.0 .. 40.0)
reset	TEXT	NOPRESET	StringRange
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state	TEXT	R/O	StringRange R/O
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.drive	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cabTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.tubeTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.rTubeBlower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.uTubeBlower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hours.unit	TEXT	R/O	StringRange R/O
hours.heater	TEXT	R/O	StringRange R/O
hours.beam	TEXT	R/O	StringRange R/O
heaterDelay	TEXT	R/O	StringRange R/O
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange DIS (1 2 1-STBY 2-STBY 3-STBY)
position.01	CHOICE		EnumRange DIS (A B)
position.02	CHOICE		EnumRange DIS (A B)
position.03	CHOICE		EnumRange DIS (A B)
position.04	CHOICE		EnumRange DIS (A B)
position.05	CHOICE		EnumRange DIS (A B)
position.06	CHOICE		EnumRange DIS (A B)
upc.enabled	CHOICE	R/O	EnumRange R/O (OFF ON)
upc.deepFade	CHOICE	R/O	EnumRange R/O (NO YES)
upc.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
upc.boost	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
upc.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
upc.fade	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
upc.deepFadeTimer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
upc.calcRatio	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
upc.rxFrequency	INTEGER		IntegerRange (7 .. 55000)
upc.clearSkyLevel	FLOAT		DoubleRange (-200.0 .. 0.0)
upc.deepFadeLevel	FLOAT		DoubleRange (-200.0 .. 0.0)
upc.deepFadeTime	INTEGER		IntegerRange (1 .. 7200)
upc.deepFadeLimit	FLOAT		DoubleRange (-200.0 .. 0.0)
upc.gain	FLOAT		DoubleRange (-9.999 .. 10.000)
upc.offset	FLOAT		DoubleRange (-200.0 .. 0.0)
upc.threshold	FLOAT		DoubleRange (0.0 .. 3.0)
upc.boostLimit	FLOAT		DoubleRange (0.0 .. 30.0)
upc.clearSkyPower	FLOAT		DoubleRange (20.0 .. 58.0)
upc.manualRatio	FLOAT		DoubleRange (0.1 .. 10.0)
upc.maxFadeSpeed	FLOAT		DoubleRange (0.1 .. 3.3)
upc.maxRecoverSpeed	FLOAT		DoubleRange (0.1 .. 3.3)
upc.txFrequency	INTEGER		IntegerRange (7 .. 55000)
upc.maxPower	FLOAT		DoubleRange (20.0 .. 58.0)
upc.input	CHOICE		EnumRange (ANALOG DIGITAL)
upc.disablePower	FLOAT		DoubleRange (20.0 .. 40.0)
upc.ratio	CHOICE		EnumRange (MANUAL CALCULATED)
upc.clearSkyNow	TEXT		StringRange
upc.clearDeepFadeTimer	TEXT		StringRange
upc.flags.boost	BOOLEAN	R/O	BooleanRange R/O (true false)
upc.flags.fade	BOOLEAN	R/O	BooleanRange R/O (true false)
upc.flags.power	BOOLEAN	R/O	BooleanRange R/O (true false)
upc.flags.beacon	BOOLEAN	R/O	BooleanRange R/O (true false)
upc.flags.attn	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
upc.primaryState	CHOICE	R/O	EnumRange R/O DIS (NONE ERROR MAINTENANCE OK)
upc.backupState	CHOICE	R/O	EnumRange R/O DIS (NONE ERROR MAINTENANCE OK)
upc.beaconUsed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
upc.primaryBeacon	INTEGER		IntegerRange DIS (1 .. 255)
upc.backupBeacon	INTEGER		IntegerRange DIS (1 .. 255)
upc.beaconFrequency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.lifeTimer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.lifeExtender	TEXT	R/O	StringRange R/O
meas.uKneeHeater	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uLifeHeater	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.lifeRemaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
runLifeExtender	TEXT	NOPRESET	StringRange
internal.days	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.seconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f10	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.f11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nomPos1	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos2	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos3	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos4	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.alc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.upc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pm	TEXT	R/O	StringRange R/O
internal.lastalc	CHOICE	R/O	EnumRange R/O (MANUAL ALC UPC)
int.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ALC limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (BUC lock warning)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low battery)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temp warning)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Tube blower over speed)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Tube over temp warning)
faults.07	ALARM	R/O	AlarmFlagRange R/O (High RF out warning)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Low RF out warning)
faults.09	ALARM	R/O	AlarmFlagRange R/O (High reflected RF warning)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RTC warn)
faults.11	ALARM	R/O	AlarmFlagRange R/O (BIPA I2C timeout)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PS I2C timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power fail sensor)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High RF drive warning)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Heater timeout warning)
faults.16	ALARM	R/O	AlarmFlagRange R/O (BIPA EEPROM)
faults.17	ALARM	R/O	AlarmFlagRange R/O (BUC lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Main controller BBRAM)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Main controller EEPROM)
faults.20	ALARM	R/O	AlarmFlagRange R/O (BIPA comm)
faults.21	ALARM	R/O	AlarmFlagRange R/O (BIPA restarted)
faults.22	ALARM	R/O	AlarmFlagRange R/O (CAN bus)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (CAN level key)
faults.24	ALARM	R/O	AlarmFlagRange R/O (PS comm)
faults.25	ALARM	R/O	AlarmFlagRange R/O (PS restarted)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF comm)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RF restarted)
faults.28	ALARM	R/O	AlarmFlagRange R/O (System)
faults.29	ALARM	R/O	AlarmFlagRange R/O (HV arc)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temp)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PS EEPROM)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Heater over voltage)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Heater under voltage)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Helix voltage detected)
faults.39	ALARM	R/O	AlarmFlagRange R/O (PS interlock)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Tube blower under speed)

name	*type*	*flags*	*range*
faults.41	ALARM	R/O	AlarmFlagRange R/O (Tube over temperature switch)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Tube over temperature)
faults.43	ALARM	R/O	AlarmFlagRange R/O (RF EEPROM)
faults.44	ALARM	R/O	AlarmFlagRange R/O (High reflected RF FLT)
faults.45	ALARM	R/O	AlarmFlagRange R/O (High RF out FLT)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Low RF out FLT)
faults.47	ALARM	R/O	AlarmFlagRange R/O (System comm)
faults.48	ALARM	R/O	AlarmFlagRange R/O (High RF drive FLT)
faults.49	ALARM	R/O	AlarmFlagRange R/O (WGSIF interlock)
faults.50	ALARM	R/O	AlarmFlagRange R/O (SEM comm)
faults.51	ALARM	R/O	AlarmFlagRange R/O (SEM restarted)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Heater timeout)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Cabinet temp sensor)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Tube temp sensor (PS))
faults.55	ALARM	R/O	AlarmFlagRange R/O (Tube temp sensor (BIPA))
faults.56	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Low main controller 24V)

name	*type*	*flags*	*range*
faults.58	ALARM	R/O	AlarmFlagRange R/O (SEM switch sys. mismatch)
faults.59	ALARM	R/O	AlarmFlagRange R/O (SIM switch sys. mismatch)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Switch system not loaded)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Local inhibit)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Ethernet CIF inhibit)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Internal inhibit)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Switch system inhibit)
faults.65	ALARM	R/O	AlarmFlagRange R/O (WGSIF inhibit)
faults.66	ALARM	R/O	AlarmFlagRange R/O (WGSIF interlock inhibit)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Serial CIF inhibit)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Web browser inhibit)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Fault recycled)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Latched fault)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Switch 1 comm)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Switch 1 stall)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Switch 2 comm)

name	*type*	*flags*	*range*
faults.74	ALARM	R/O	AlarmFlagRange R/O (Switch 2 stall)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Switch 3 comm)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Switch 3 stall)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Switch 4 comm)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Switch 4 stall)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Switch 5 comm)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Switch 5 stall)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Switch 6 comm)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Switch 6 stall)
faults.89	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.79	ALARM	R/O	AlarmFlagRange R/O (UPC Boost Limit)
faults.80	ALARM	R/O	AlarmFlagRange R/O (UPC Fade Limit)
faults.81	ALARM	R/O	AlarmFlagRange R/O (UPC Power Limit)
faults.82	ALARM	R/O	AlarmFlagRange R/O (UPC Beacon Status)
faults.83	ALARM	R/O	AlarmFlagRange R/O (UPC Attn Limit)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.91	ALARM	R/O	AlarmFlagRange R/O (Fast HV shutdown)
faults.92	ALARM	R/O	AlarmFlagRange R/O (Sim. active node)
faults.93	ALARM	R/O	AlarmFlagRange R/O (Cntlr power shutdown)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Ext interlock inhibit)
faults.95	ALARM	R/O	AlarmFlagRange R/O (BCM restarted)
faults.96	ALARM	R/O	AlarmFlagRange R/O (BCM Comm)
faults.97	ALARM	R/O	AlarmFlagRange R/O (BCM EEPROM)
faults.98	ALARM	R/O	AlarmFlagRange R/O (Life extender refresh req.)
faults.100	ALARM	R/O	AlarmFlagRange R/O (Low TWT life)
faults.101	ALARM	R/O	AlarmFlagRange R/O (RF limiter)
faults.102	ALARM	R/O	AlarmFlagRange R/O (Beam over current)
faults.103	ALARM	R/O	AlarmFlagRange R/O (Beam under current)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Tube fan under speed)
faults.126	ALARM	R/O	AlarmFlagRange R/O (Beam/HV Off)

9.89 CPI-SLPA

This page describes the device driver and the device window for the CPI SuperLinear TWTA series with support for redundancy and up to 8 waveguide switches

The driver supports the following models:

- TL22CI

- TL12UI

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
control.Point	Defines the control via serial or ethernet interface.
redundancy	Defines the redundancy mode
loRfOutAlarm	The low output power alarm trip point.
hiRfOutAlarm	The high output power alarm trip point.
loRfOutFault	The low output power fault trip point.
hiRfOutFault	The high output power fault trip point.
Serial	Remember to set the line parameters for the serial interface. of the HPA is *9600,E,7,1* .
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" in the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	*type*	*flags*	*range*

9.90 CPI-SRCU-BCIP

This page describes the device driver and the device window for the CPI SRCU redundancy controller using binary protocol BCIP.

It was tested with 1:1 Red Model: B5KO (160W Ka-Band) GaN SSPAs

- FW (SRCU): 03.00.22 / Boot 01.01.05
- FW (SSPA): 01.03.24 / Boot 01.00.00

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*

parameter	*description*
address	The device address (decimal). The factory default is 48.
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
redundancy	Defines the redundancy type.
switches	Defines the number of installed switches 0-16.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" int the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1)
config.switches	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange (1 2)
position.01	CHOICE		EnumRange DIS (A B)
position.02	CHOICE		EnumRange DIS (A B)
position.03	CHOICE		EnumRange DIS (A B)
position.04	CHOICE		EnumRange DIS (A B)
position.05	CHOICE		EnumRange DIS (A B)
position.06	CHOICE		EnumRange DIS (A B)
position.07	CHOICE		EnumRange DIS (A B)
position.08	CHOICE		EnumRange DIS (A B)
position.09	CHOICE		EnumRange DIS (A B)
position.10	CHOICE		EnumRange DIS (A B)
position.11	CHOICE		EnumRange DIS (A B)
position.12	CHOICE		EnumRange DIS (A B)
position.13	CHOICE		EnumRange DIS (A B)
position.14	CHOICE		EnumRange DIS (A B)
position.15	CHOICE		EnumRange DIS (A B)
position.16	CHOICE		EnumRange DIS (A B)
internal.sw0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.sw3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Switch 1 comm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch 1 stall)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch 2 comm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch 2 stall)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch 3 comm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch 3 stall)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Switch 4 comm)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch 4 stall)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch 5 comm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch 5 stall)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch 6 comm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Switch 6 stall)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Switch 7 comm)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switch 7 stall)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch 8 comm)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switch 8 stall)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch 9 comm)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (Switch 9 stall)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Switch 10 comm)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Switch 10 stall)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 11 comm)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 11 stall)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 12 comm)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 12 stall)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch 13 comm)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch 13 stall)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch 14 comm)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch 14 stall)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch 15 comm)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch 15 stall)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch 16 comm)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch 16 stall)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.91 CPI-SRCU

This page describes the device driver and the device window for the CPI SRCU redundancy controller using CPI-STX protocol.

It was tested with 1:1 Redundancy Compact CMPA

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal). The factory default is 65.
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
redundancy	Defines the redundancy type.
switches	Defines the number of installed switches 0-16.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" int the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	CHOICE	R/O	EnumRange R/O (N/A 1:1 1:1-MAINTENANCE SWITCH 1:1-PHASE COMBINED 1:2 1:2-MAINTENANCE SWITCH 1:2-PHASE COMBINED)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
location	CHOICE	R/O	EnumRange R/O (1 2)
position.01	CHOICE		EnumRange (A B)
position.02	CHOICE		EnumRange (A B)
position.03	CHOICE		EnumRange (A B)
position.04	CHOICE		EnumRange (A B)
internal.test	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HPA 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (HPA 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (HPA 3)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)

9.92 CPI-Switch

This page describes the device driver and the device window for the CPI Microwave Waveguide Switching and HPA Protection system tested with 3 waveguide switches and 3+1 HPA redundancy

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
reset	TEXT	NOPRESET	StringRange
switch.01	CHOICE		EnumRange DIS (A B FLT)
switch.02	CHOICE		EnumRange DIS (A B FLT)
switch.03	CHOICE		EnumRange DIS (A B FLT)
switch.04	CHOICE		EnumRange DIS (A B FLT)
switch.05	CHOICE		EnumRange DIS (A B FLT)
switch.06	CHOICE		EnumRange DIS (A B FLT)
switch.07	CHOICE		EnumRange DIS (A B FLT)
switch.08	CHOICE		EnumRange DIS (A B FLT)
switch.09	CHOICE		EnumRange DIS (A B FLT)
switch.10	CHOICE		EnumRange DIS (A B FLT)
switch.11	CHOICE		EnumRange DIS (A B FLT)
switch.12	CHOICE		EnumRange DIS (A B FLT)
internal.f21	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f22	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f23	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f24	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f25	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f26	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f27	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f28	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f29	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f30	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.f31	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.f32	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ext. interlock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unknown model)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HPA 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (HPA 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (HPA 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (HPA 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (HPA 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (HPA 6)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch 8)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch 9)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch 12)

9.93 CPI-TWTA-Ka

This page describes the device driver and the device window for the CPI TWT Ka/K-Band

amplifiers. Tested with Ka-Band Outdoor HPA TWT Ka 250W CW. This driver does not support ALC and connected waveguide switches.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* or if you use the TCP interface use port number 50000.
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.standbyElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.transmitElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.unitOnElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.checkLimits	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.tubeTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (SSIPA Over Current)
faults.04	ALARM	R/O	AlarmFlagRange R/O (BUC Alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Tube Over Temp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High voltage arc)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC buss)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Helix voltage in standby)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High Reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PS interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Tube switch over temp)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (External interlock)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Tube over temp)
faults.21	ALARM	R/O	AlarmFlagRange R/O (SSIPA over current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (BUC fault)
faults.23	ALARM	R/O	AlarmFlagRange R/O (BBRAM data invalid)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Internal reflected RF)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Fan stalled)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Isolator over temp)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temp)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Power sensor)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.94 CPI-TWTA

This page describes the device driver and the device window for the CPI TWT amplifiers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.checkLimits	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 25.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 25.0)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (BB RAM)
faults.04	ALARM	R/O	AlarmFlagRange R/O (BUC unlock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power fail timeout)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC buss under voltage)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Helix voltage in standby)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.21	ALARM	R/O	AlarmFlagRange R/O (SSIPA over current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.95 CPI-TouchPower

This page describes the device driver and the device window for the CPI Indoor TWTA with touch screen (TouchPower series)

The driver was tested with the following models:

- T5UI 750W Ku-Band TWTA

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- Displays the redundancy settings
- --- Displays Uplink Power Control parameters.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
loRfOutAlarm	The low output power alarm trip point.
hiRfOutAlarm	The high output power alarm trip point.
loRfOutFault	The low output power fault trip point.
hiRfOutFault	The high output power fault trip point.
config.hiRefOutAlarm	The high reflected power alarm trip point.
config.hiRefOutFault	The high reflected power fault trip point.
config.hiRfInpAlarm	The high input power alarm trip point.
config.hiRfInpFault	The high input power fault trip point.
redundancy	Defines the redundancy mode.
switches	Defines the number of switches.
nominalPosSw1	Defines the nominal position of switch 1 in a 2:1 redundancy.
nominalPosSw2	Defines the nominal position of switch 2 in a 2:1 redundancy.
nominalPosSw3	Defines the nominal position of switch 3 in a 2:1 redundancy.
nominalPosSw4	Defines the nominal position of switch 4 in a 2:1 redundancy.
upcExtInhibit	Enables or disables the beacon receiver fail alarm indication.
Serial	Remember to set the line parameters for the serial interface. of the HPA is *9600,E,7,1* .
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0

parameter	*description*
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet Interface

The ODUs may be controlled through their Ethernet interface. The port number for this is 50000. Assuming the amplifier is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:50000" in the sat-nms VLC setup in order to make the VLC talk to the amplifier. Beside this, the amplifier must be configured to "Ethernet CIF" at it's web-interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.amptype	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.main.kernel.rev	TEXT	R/O	StringRange R/O
info.main.panel.rev	TEXT	R/O	StringRange R/O
info.switchSystem	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
info.swSystemAmpId	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
info.hasUPC	CHOICE	R/O	EnumRange R/O (NO YES)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.loRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRefOutAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRefOutFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfInpAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.hiRfInpFault	FLOAT	SETUP	DoubleRange (0.0 .. 2000.0)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (STANDALONE 1:1 2:1)
config.switches	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6)
config.upcExtInhibit	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.nominalPosSw1	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw2	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw3	CHOICE	SAVE SETUP	EnumRange (A B)
config.nominalPosSw4	CHOICE	SAVE SETUP	EnumRange (A B)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.alc	CHOICE		EnumRange (MANUAL ALC UPC)
tx.requestedLevel	FLOAT		DoubleRange (-100.0 .. 40.0)
reset	TEXT	NOPRESET	StringRange
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state	TEXT	R/O	StringRange R/O
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.drive	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cabTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.rTubeBlower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hours.unit	TEXT	R/O	StringRange R/O
hours.heater	TEXT	R/O	StringRange R/O
hours.beam	TEXT	R/O	StringRange R/O
heaterDelay	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
location	CHOICE		EnumRange DIS (1 2 1-STBY 2-STBY 3-STBY)
position.01	CHOICE		EnumRange DIS (A B)
position.02	CHOICE		EnumRange DIS (A B)
position.03	CHOICE		EnumRange DIS (A B)
position.04	CHOICE		EnumRange DIS (A B)
position.05	CHOICE		EnumRange DIS (A B)
position.06	CHOICE		EnumRange DIS (A B)
upc.enabled	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
upc.deepFade	CHOICE	R/O	EnumRange R/O DIS (NO YES)
upc.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.boost	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.voltage	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
upc.fade	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.deepFadeTimer	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
upc.calcRatio	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
upc.rxFrequency	INTEGER		IntegerRange DIS (7 .. 55000)
upc.clearSkyLevel	FLOAT		DoubleRange DIS (-200.0 .. 0.0)
upc.deepFadeLevel	FLOAT		DoubleRange DIS (-200.0 .. 0.0)
upc.deepFadeTime	INTEGER		IntegerRange DIS (1 .. 7200)
upc.deepFadeLimit	FLOAT		DoubleRange DIS (-200.0 .. 0.0)
upc.gain	FLOAT		DoubleRange DIS (-9.999 .. 10.000)
upc.offset	FLOAT		DoubleRange DIS (-200.0 .. 0.0)
upc.threshold	FLOAT		DoubleRange DIS (0.0 .. 3.0)

name	*type*	*flags*	*range*
upc.boostLimit	FLOAT		DoubleRange DIS (0.0 .. 30.0)
upc.clearSkyPower	FLOAT		DoubleRange DIS (20.0 .. 58.0)
upc.manualRatio	FLOAT		DoubleRange DIS (0.1 .. 10.0)
upc.maxFadeSpeed	FLOAT		DoubleRange DIS (0.1 .. 3.3)
upc.maxRecoverSpeed	FLOAT		DoubleRange DIS (0.1 .. 3.3)
upc.txFrequency	INTEGER		IntegerRange DIS (7 .. 55000)
upc.maxPower	FLOAT		DoubleRange DIS (20.0 .. 58.0)
upc.input	CHOICE		EnumRange DIS (ANALOG DIGITAL)
upc.disablePower	FLOAT		DoubleRange DIS (20.0 .. 40.0)
upc.ratio	CHOICE		EnumRange DIS (MANUAL CALCULATED)
upc.clearSkyNow	TEXT		StringRange DIS
upc.clearDeepFadeTimer	TEXT		StringRange DIS
upc.flags.boost	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.flags.fade	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.flags.power	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.flags.beacon	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.flags.attn	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.primaryState	CHOICE	R/O	EnumRange R/O DIS (NONE ERROR MAINTENANCE OK)
upc.backupState	CHOICE	R/O	EnumRange R/O DIS (NONE ERROR MAINTENANCE OK)
upc.beaconUsed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
upc.primaryBeacon	INTEGER		IntegerRange DIS (1 .. 255)
upc.backupBeacon	INTEGER		IntegerRange DIS (1 .. 255)
upc.beaconFrequency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
internal.days	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.seconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loc2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nomPos1	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos2	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos3	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.nomPos4	INTEGER	R/O	IntegerRange R/O (1 .. 2)
internal.alc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.upc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pm	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.lastalc	CHOICE	R/O	EnumRange R/O (MANUAL ALC UPC)
int.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ALC limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (BUC lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low battery)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Tube blower over speed)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Tube over temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (High RF out ALM)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Low RF out ALM)
faults.09	ALARM	R/O	AlarmFlagRange R/O (High reflected RF ALM)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RTC)
faults.11	ALARM	R/O	AlarmFlagRange R/O (BIPA I2C timeout)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PS I2C timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power fail sensor)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High RF drive ALM)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Heater timeout)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (BIPA EEPROM)
faults.17	ALARM	R/O	AlarmFlagRange R/O (BUC lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Main controller BBRAM)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Main controller EEPROM)
faults.20	ALARM	R/O	AlarmFlagRange R/O (BIPA comm)
faults.21	ALARM	R/O	AlarmFlagRange R/O (BIPA restarted)
faults.22	ALARM	R/O	AlarmFlagRange R/O (CAN bus)
faults.23	ALARM	R/O	AlarmFlagRange R/O (CAN level key)
faults.24	ALARM	R/O	AlarmFlagRange R/O (PS comm)
faults.25	ALARM	R/O	AlarmFlagRange R/O (PS restarted)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF comm)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RF restarted)
faults.28	ALARM	R/O	AlarmFlagRange R/O (System)
faults.29	ALARM	R/O	AlarmFlagRange R/O (HV arc)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Cabinet over temp)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PS EEPROM)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Heater over voltage)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Heater under voltage)

name	*type*	*flags*	*range*
faults.35	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Helix over voltage)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Helix under voltage)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Helix voltage detected)
faults.39	ALARM	R/O	AlarmFlagRange R/O (PS interlock)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Tube blower under speed)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Tube over temperature)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Tube over temperature switch)
faults.43	ALARM	R/O	AlarmFlagRange R/O (RF EEPROM)
faults.44	ALARM	R/O	AlarmFlagRange R/O (High reflected RF FLT)
faults.45	ALARM	R/O	AlarmFlagRange R/O (High RF out FLT)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Low RF out FLT)
faults.47	ALARM	R/O	AlarmFlagRange R/O (System comm)
faults.48	ALARM	R/O	AlarmFlagRange R/O (High RF drive FLT)
faults.49	ALARM	R/O	AlarmFlagRange R/O (WGSIF interlock)
faults.50	ALARM	R/O	AlarmFlagRange R/O (SEM comm)

name	*type*	*flags*	*range*
faults.51	ALARM	R/O	AlarmFlagRange R/O (SEM restarted)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Heater timeout)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Cabinet temp sensor)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Tube temp sensor (PS))
faults.55	ALARM	R/O	AlarmFlagRange R/O (Tube temp sensor (BIPA))
faults.56	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Low main controller 24V)
faults.58	ALARM	R/O	AlarmFlagRange R/O (SEM switch system mismatch)
faults.59	ALARM	R/O	AlarmFlagRange R/O (SIM switch system mismatch)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Switch system not loaded)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Local inhibit)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Ethernet CIF inhibit)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Internal inhibit)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Switch system inhibit)
faults.65	ALARM	R/O	AlarmFlagRange R/O (WGSIF inhibit)
faults.66	ALARM	R/O	AlarmFlagRange R/O (WGSIF interlock inhibit)

name	*type*	*flags*	*range*
faults.67	ALARM	R/O	AlarmFlagRange R/O (Serial CIF inhibit)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Web browser inhibit)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Fault recycled)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Latched fault)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Switch 1 comm)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Switch 1 stall)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Switch 2 comm)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Switch 2 stall)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Switch 3 comm)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Switch 3 stall)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Switch 4 comm)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Switch 4 stall)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Switch 5 comm)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Switch 5 stall)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Switch 6 comm)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Switch 6 stall)

name	*type*	*flags*	*range*
faults.89	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.79	ALARM	R/O	AlarmFlagRange R/O (UPC Boost Limit)
faults.80	ALARM	R/O	AlarmFlagRange R/O (UPC Fade Limit)
faults.81	ALARM	R/O	AlarmFlagRange R/O (UPC Power Limit)
faults.82	ALARM	R/O	AlarmFlagRange R/O (UPC Beacon Status)
faults.83	ALARM	R/O	AlarmFlagRange R/O (UPC Attn Limit)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.96 CPI-VZU-TWTA

This page describes the device driver and the device window for the CPI VZU TWT amplifiers. The driver has been tested with the following model: VZU6995F (via CIM VJW2740092)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*
address	The device address (decimal). While communicating via CIM the first HPA's address is 65 (A) and the second HPA's address is 66 (B).

parameter	*description*

Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.summary	TEXT	R/O	StringRange R/O
info.s1	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.beam	CHOICE		EnumRange (STANDBY ON)
reset	TEXT	NOPRESET	StringRange
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 80.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.summ1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.summ2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.beam	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (High reflected power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power peak indicate)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Heater not ready)

9.97 CSM-Spectrum-Analyzer

This page describes the CSM-Spectrum-Analyzer device driver and the device window for this device.

The CSM-Spectrum-Analyzer device driver provides some sort of bridge which makes the spectrum analyzer device drivers contained in the sat-nms CSM software available as M&C/VLC devices. Using this type of device driver is the recommended way to connect a spectrum analyzer to M&C/VLC.

Actually there are five types of spectrum analyzers supported:

spectrum analyzer	*CSM driver*	*supported interfaces*
Agilent ESA E44xx spectrum analyzers	AgilentSpectrumAnalyzer	GPIB or RS-232
Advantec spectrum analyzers	AdvantecSpectrumAnalyzer	GPIB or RS-232
Rohde & Schwarz FSC-3 spectrum analyzer	FscSpectrumAnalyzer	NI-VISA (TCPIP / VXI-11)
Rohde & Schwarz FSL-3 spectrum analyzer	Fsl3SpectrumAnalyzer	NI-VISA (TCPIP / VXI-11)
Rohde & Schwarz FSV-7 spectrum analyzer	FsvSpectrumAnalyzer	NI-VISA (TCPIP / VXI-11)
Rohde & Schwarz FSH-4 handheld spectrum analyzer	FshSpectrumAnalyzer	NI-VISA (TCPIP / VXI-11)

spectrum analyzer	***CSM driver***	***supported interfaces***
Rohde & Schwarz Spectrum Rider FPH handheld spectrum analyzer	FphSpectrumAnalyzer	NI-VISA (TCPIP / VXI-11)
LP Technologies LPT3000 spectrum analyzer		
(for this device you find additional inormation at the end of this page)	LPT3000SpectrumAnalyzer	Serial via TCP/IP
Narda NPA 2500/3000/6000 spectrum analyzers	NardaSpectrumAnalyzer	Serial via TCP/IP
SED Calian Decimator D3 / D4 spectrum analyzers	DecimatorD4SpectrumAnalyzer	Serial via TCP/IP

When using this driver, assigning and configuring the physical interface to the spectrum analyzer is done slightly different to other devices. Please read the paragraph "How to configure the driver" at the and of this chapter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- This page gives you access to the first 16 instrument memories of the spectrum analyzer. Clicking to 'STO x' stores the actual instrument settings to this memory place in the device. Clicking to 'RCL x' recalls the selected memory.You may use the text field between the buttons to note a comment for this particular memory place. Unless the instrument settings which are stored in the device itself, the text is stored at the VLC/M&C system. Hence, if you replace the spectrum analyzer, the stored instrument settings are gone with the replaced device, the comments however are still present.Also this page is available as a separate window. click to 'PRE' at the upper left corner of the spectrum view to launch a 'preset' window without leaving the spectrum view.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

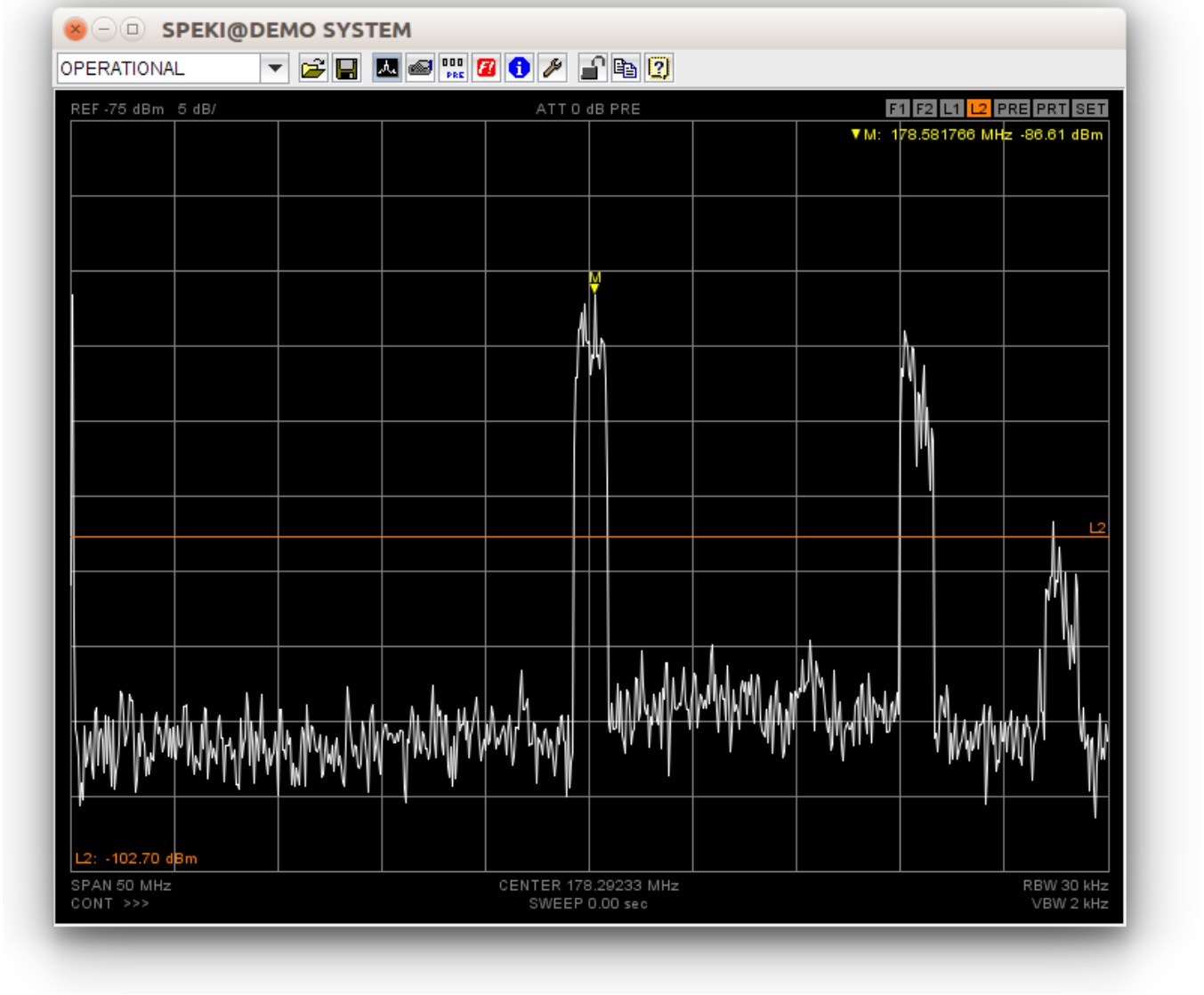
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The CSM interface / device address string. See the paragraph "CSM address specification" at the and of this chapter for details.
display	This configuration parameter switches the local display of the spectrum analyzer on or off. Switching off the display improves the measurement speed of the analyzer in some situations. Please note, that some spectrum analyzers may ignore this parameter.
dispTimeout	If you set this parameter to a non-zero value, the software will switch off the device's display automatically after the given number of minutes.
model	Select the type of spectrum analyzer you want to control.
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature. See chapter 'Fixed LO frequency' below if you have no input switch but you want to translate the frequency by a fixed LO frequency.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'. See chapter 'Fixed LO frequency' below if you have no input switch but you want to translate the frequency by a fixed LO frequency.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page.
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

Display lines

The spectrum display provides four display lines, two horizontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implemented in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To access the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of sweeps already averaged.

These display functions are implemented locally in the spectrum display window and therefore not visible at the spectrum analyzers 'physical' display. The setting which control the display mode however is a common parameter of the spectrum analyzer driver. As a consequence, changing the display mode parameter changes the display at all places where the display of this spectrum analyzer is watched. Also the display mode setting may be stored as a part of a

device preset.

Preset handling

Settings of a spectrum analyzer may be stored in two different ways:

The fist method is to store the device setting in one of the memories in the device itself. The paragraph "Display window pages" above describes how to do this. The advantage of this method is, that the spectrum analyzer stores **all** it's settings and also restores all of them when the preset gets recalled. The disadvantage if this method is, that cannot apply such a stored setup to another device.

Beside this, the csm spectrum analyzer driver provides the standard sat-nms device preset method in a slightly modified way, optimized for a spectrum analyzer's behaviour. If you store a device preset, all those "coupled functions" parameters are skipped which are marked as AUTO. When the preset gets applied, such a parameter is not explicitly set, the spectrum analyzer itself chooses the proper value for this parameter. Additionally the parameters of a device preset are applied in a sequence which corresponds to the dependencies of "coupled functions" parameters.

The advantage of this method is, that sat-nms device presets are stored at a central place, a preset recorded with one device may be applied to any other one. As the csm spectrum analyzer driver only implements a "standard" subset of spectrum analyzer parameters, recorded presets are mostly compatible between devices from different vendors. However, you should be aware that applying a sat-nms device preset only sets a couple of parameters at the device, it does not recall a complete setup like the internal memory function of the spectrum analyzer does. In some cases it may be necessary to apply a "INSTR PRESET" first, to initialize the spectrum analyzer to a defined state.

How to configure the driver

As mentioned above, the CSM-Spectrum-Analyzer device driver uses the interface handling of the sat-nms CSM software to access the spectrum analyzer. From the VLC's point of view, the driver behaves like a 'logical' device, without using any physical interface. In practice, configuring a CSM-Spectrum-Analyzer device is done as follows:

1. In the device setup window define a 'null' interface running the 'Logical' protocol for each spectrum analyzer you want to control. For best performance, each device should be assigned to an interface of it's own. To achieve a fast display update, you should tune the idle time setting. The recommended idle time depends on the speed of the client-server network connection:

network connection	***idle time***
Ethernet	100 msec
ISDN 64kbps	1000 msec

network connection	***idle time***
Modem / analog telephone line	2000 msec

Larger idle times slow down the maximum sweep refresh rate of the driver, thus minimizing the risk of buffer overflows through slow communication lines. If the spectrum analyzer seems to respond to parameter changes after only some sweeps delayed, you should increase the idle time. The same applies if the client server connection breaks after some time. 2. At the maintenance page / setup dialog, select the type of spectrum analyzer to control. 3. At the 'address' field, type in the interface / address specification.

Fixed LO frequency

At the maintenance / setup page you can configure the driver to control an input switch in front of the spectrum analyzer and an individual LO frequency for each switch position.

If there is no input switch, but nevertheless you want to translate the frequency reading, you may 'fake' an input switch with one single position.

Example: The Antenna receives Ku-band, the spectrum analyzer is connected beyond an LNB with an 10GHz LO. To make the spectrum analyzer show the center / marker frequencies in terms of Ku-band rather than L-band, configure the input switch parameters as follows:

- inputSelId: --- SP.info.port
- inputNames: --- null
- inputLoFreqs: --- 10000

The example above assumes, that the device name of the spectrum analyzer is 'SP'. It uses it's communication port description as a dummy input switch parameter ID. With a CSM spectrum analyzer device this always reads 'null', hence always the 10000MHz LO frequency are applied as specified in the 'inputLoFreqs' field.

CSM address specification

The CSM software (and therefore the CSM-Spectrum-Analyzer driver) knows three major interface types to control a device. They are:

- RS232 --- Serial interfaces are provided by the Advantest spectrum analyzer and by some models of the Agilent ESA units. The disadvantage of this common interface type is the comparable low speed. Even with 100 msecs idle time the best sweep rate that can be achieved is about 2 sweeps per second.**ttyS1:19200,8,N,1**For a serial interface, the (Linux-) device name of the serial interface, followed by a colon and the interface settings have to be filled into the 'address' field of the setup dialog (like the example above). When using a terminal server which is recommended for baud rates above 19200, the address field must contain the IP address of the terminal server, followed by a colon and the port number. Example:**192.168.100.100:4001**
- GPIB --- The sat-nms software supports the use of GPIB with the ENET100 network to GPIB interface by National Instruments and with the Prologix Ethernet to GPIB controller. The

address specification for a GPIB controlled device looks like this (for the ENET100):**gpib0:13**The keyword 'gpib0' addresses the first ENET100 interface known to the VLC. The number following the colon character denotes the GPIB address of the device.With the Prologix GPIP adapter the address specification looks like this:**gpibx:10.0.0.1:13**'gpibx' tells the software that a Prologix adapter is used, 10.0.0.1 is the IP address of the adapter. Finally, 13 is the GPIB address of the spectrum analyzer as in the example above.

- NI-VISA --- The Rohde & Schwarz FLS-3 spectrum analyzer is controlled using the VISA instrument control library by National Instruments. The sat-nms software uses a program called visaproxy to access the spectrum analyzer. The visaproxy must be configured as described at the end of this chapter. The address specification for a visaproxy device looks like this:**visaproxy:localhost:5000**The keyword visaproxy tells the software about this type of interface. The second field designates the host name or IP address where the visaproxy service runs (usually *localhost*). Please note that this address is*not* the IP address of the spectrum analyzer. The latter is set at the visaproxy configuration as described below. The last field defines the port number at which the visaproxy listens. Out of the box, the sat-nms software uses port 5000 for the first spectrum analyzer, 5001 for the second, and so on.

The visaproxy configuration

When controlling NI-VISA devices, the sat-nms software does this through a network service called *visaproxy* . Before the sat-nms software can access a spectrum analyzer through this service, it must be configured with the device address.

This is done by editing the file 'visaproxy.setup' in the VLC's home directory. With a sat-nms system out of the box this file looks like this:

```
#
# /home/satnms/visaproxy.setup
#
# Example for the contents of the visaproxy configuration file.
# The file must be copied to the location mentioned above.
#
# The configuration file is read by the visaproxy startup script,
# defines which visaproxy instances are launches at system start.
# It must define two shell variables:
#
# PORTNO      The IP port number of the first instance that is
#             started. Subsequent instances each increment the
#             port number by 1
#
# INSTRUMENTS The VISA resource descriptions if the instruments
#             to serve, separated by space characters. Actually
#             the sat-nms software only supports TCPIP / VXI-11
#             based VISA instruments, hence the resource
#             descriptions all look like TCPIP::aaa.bbb.ccc.ddd
#
# The example below defines two instances of the visaproxy,
# the first proxy makes the instrument TCPIP::192.168.1.1
# accessible at port 5000, the second one instrument
# TCPIP::192.168.1.2 at port 5001
#
#
# PORTNO=5000
# INSTRMENTS="TCPIP::192.168.1.1 TCPIP::192.168.1.2"
```

To activate the visaproxy network service, remove the comment characters in front of the last two lines of the file and edit the INSTRUMENTS list of spectrum analyzer addresses for your need. Normally there is no need to change the PORTNO definition. Then save the file and re-boot the VLC. After starting up, the VLC will run a visaproxy service for each of the configured addresses.

Troubleshooting a GPIB ENET100 interface

The ENET100 GPIB interface has been reported to block the communication to the spectrum analyzer under certain conditions. To fix this problem, perform a system restart with the following sequence of operations:

1. Shut down the M&C computer.
2. Switch off the spectrum analyzer and the ENET100 interface box.
3. Switch on the spectrum analyzer.
4. Switch on the ENET100 interface box.
5. Finally switch on the M&C computer again.

After booted up, the sat-nms software now should recognize the spectrum analyzer again.

LP Technologies LPT3000

The LPT3000 is an exclusively remote controlled spectrum analyzer with no internal display. Hence the configuration parameters concerning the local display are not applicable with this device.

The CSM device driver controls the LPT3000 using TCP/IP. You have to setup the address field as described above for a RS232 terminal server. For a LPT3000 at the IP address 10.0.0.1 the address field would be "10.0.0.1:5000", the LPT3000 always listens at port 5000.

Unlike other spectrum analyzers, the LPT3000 provides an input selection switch (4:1) which can be controlled through the sat-nms software. The 'input' Parameter of the device reflects the actual input selection (1 .. 4), the parameter also may be used to control the input selection. To configure this for a spectrum analyzer device called "SPA", setup the configuration variables at the SPA device as follows:

1. 'inputSelId' must be set to 'SPA.input', this makes the spectrum analyzer driver watch it's own input selector.
2. 'inputNames' must be set to '1,2,3,4' which are the possible input selections.
3. 'inputLoFreqs' must be set to the list of LO frequencies applying to the inputs 1 .. 4 (in that order).

You may want to control the input selector with user defined names for the switch position. To achieve this, set up a Choice-Translator device to do the name translation.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.pdfilter	TEXT	R/O	StringRange R/O
config.display	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.dispTimeout	INTEGER	SAVE SETUP	IntegerRange (0 .. 60)
config.model	CHOICE	SAVE SETUP	EnumRange (FscSpectrumAnalyzer Fsl3SpectrumAnalyzer FsvSpectrumAnalyzer FshSpectrumAnalyzer FphSpectrumAnalyzer AgilentSpectrumAnalyzer AdvantestSpectrumAnalyzer LPT3000SpectrumAnalyzer LPT3000RX4SpectrumAnalyzer NardaSpectrumAnalyzer DecimatorD4SpectrumAnalyzer)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
address	TEXT	SAVE	StringRange
internal.tick	TEXT	R/O	StringRange R/O
title	TEXT	R/O	StringRange R/O
input	INTEGER		IntegerRange (0 .. 0)
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.004 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
memname.0	TEXT	SAVE	StringRange
memname.1	TEXT	SAVE	StringRange
memname.2	TEXT	SAVE	StringRange
memname.3	TEXT	SAVE	StringRange
memname.4	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
memname.5	TEXT	SAVE	StringRange
memname.6	TEXT	SAVE	StringRange
memname.7	TEXT	SAVE	StringRange
memname.8	TEXT	SAVE	StringRange
memname.9	TEXT	SAVE	StringRange
memname.10	TEXT	SAVE	StringRange
memname.11	TEXT	SAVE	StringRange
memname.12	TEXT	SAVE	StringRange
memname.13	TEXT	SAVE	StringRange
memname.14	TEXT	SAVE	StringRange
memname.15	TEXT	SAVE	StringRange
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

9.98 Calian-Decimator

This page describes the Calian-Decimator device driver and the device window for this device. The driver supports Decimator D3 and D4 spectrum analyzers by Calian.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

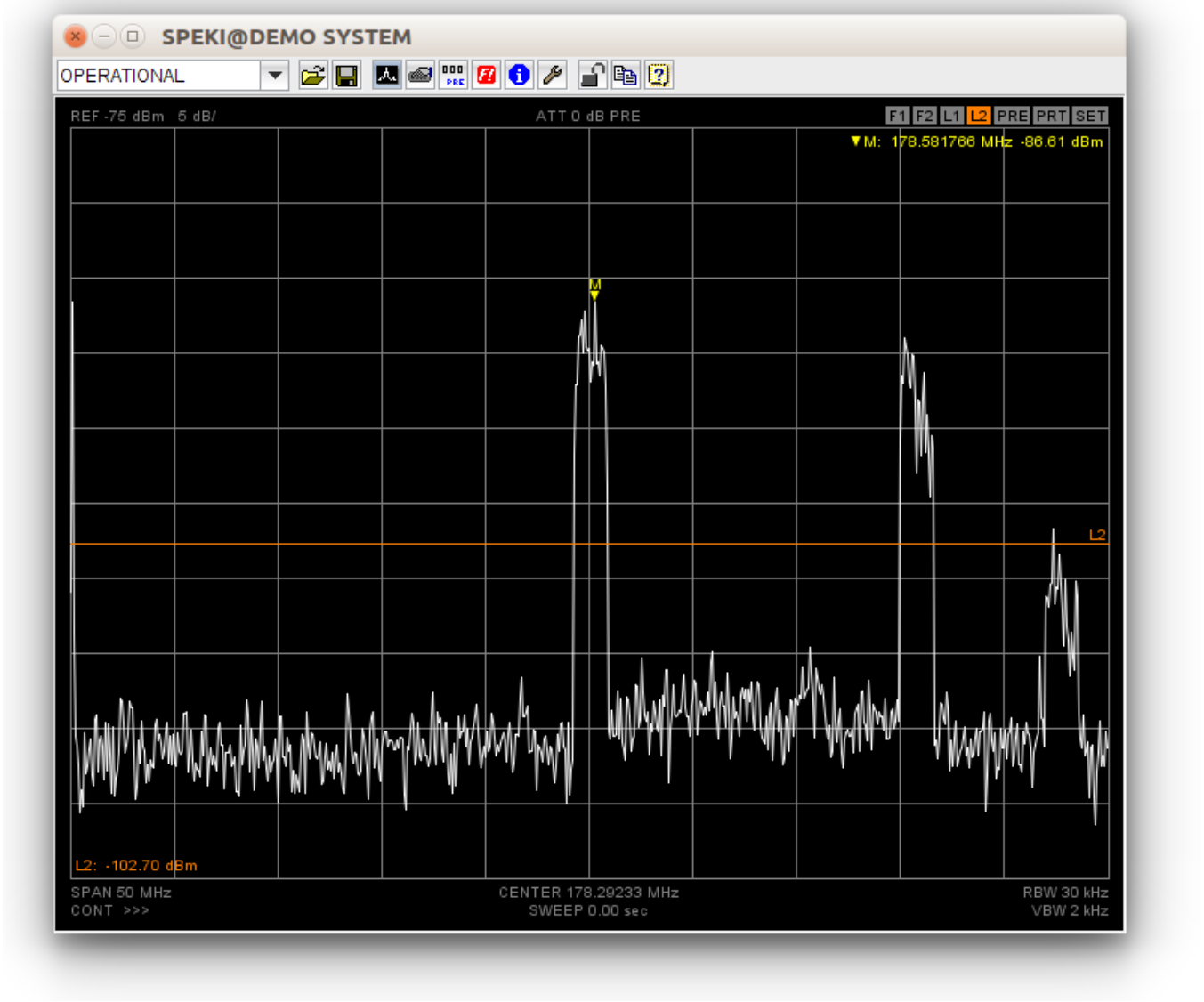
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page (not available with the D3/D4 spectrum analyzers)
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The D3/D4 spectrum analyzers black all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG

mode is followed by the number of swees already averaged.

Preset handling

The Calian-Decimator driver provides the standard sat-nms device preset method in a slightly modified way, optimized for a spectrum analyzer's behaviour. If you store a device preset, all those "coupled functions" parameters are skipped which are marked as AUTO. When the preset gets applied, such a parameter is not explicitely set, the spectrum analyzer itself chooses the proper value for this parameter. Additionally the parameters of a device preset are applied in a sequence which corresponds to the dependencies of "coupled functions" parameters.

The advantage of this method is, that sat-nms device presets are stored at a central place, a preset recorded with one device may be applied to any other one. As the spectrum analyzer driver only implements a "standard" subset of spectrum analyzer parameters, recorded presets are mostly compatible between devices from different vendors. However, you should be aware that applying a sat-nms device preset only sets a couple of parameters at the device, it does not recall a complete setup like the internal memory function of the spectrum analyzer does. In some cases it may be necessary to apply a "INSTR PRESET" first, to initialize the spectrum analyzer to a defined state.

How to configure the driver

The Decimator D3/D4 spectrum analyzers are controlled via a TCP connection at port 9784. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 9784. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwrevision	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.pdfilter	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0 .. 0)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
title	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange DIS (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT	R/O	DoubleRange R/O (0.004 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)

name	*type*	*flags*	*range*
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE	R/O	EnumRange R/O (AUTO)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
pdfilter	CHOICE	SAVE	EnumRange (VBW AVG)
avg	INTEGER	SAVE	IntegerRange (1 .. 255)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange DIS
recall	TEXT	NOPRESET	StringRange DIS
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT		DoubleRange (0.000000 .. 0.000000)
lband.marker.x	FLOAT		DoubleRange (0.000000 .. 0.000000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Overload)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Mixer cal.)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Basebans cal.)

9.99 Calian-MCI-ACU

This page describes the device driver and the device window for the Calian MCI ACU. The driver monitors the ACU's state and position and permits to point the antenna to given azimuth/elevation angles. TLE tracking is also supported, thei requires that the ACU's geographic location ist set at the ACU and NTP time synchronisatio has been successfully set up.

The ACU is controlled via Modbus/TCP. It expects the M&C to connect at IP address of the azimuth MCI, port 502. The Modbus unit address must be set to '3'. In the M&C device configuration an IP based serial communication interfacce at the ACU's IP adrress / port number id used, the Modbus unit address is set in the 'address' field of the device setup window.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the TLE parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The Modbus unit address. Must be set to '3'
az.lolimit	The azimuth low (ccw) soft limit
az.hilimit	The azimuth high (cw) soft limit
el.lolimit	The elevation low softlimit
el.hilimit	The elevation high softlimit

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.runmode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.trackmode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.bcrx.present	TEXT	R/O	StringRange R/O
internal.statusWord	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.posreach.az	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.posreach.el	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.posreach.tl	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.parking	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.x	INTEGER	R/O	IntegerRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.0000 .. 0.0000)
elevation	FLOAT		DoubleRange (0.0000 .. 0.0000)
polarization	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.el	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.tl	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.pl	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
positionReached	BOOLEAN	R/O	BooleanRange R/O (true false)
azLimitProximity	TEXT	R/O	StringRange R/O
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
motorStop	TEXT	NOPRESET	StringRange
trackTLE	TEXT	NOPRESET	StringRange
trackSTEP	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange
service	TEXT	NOPRESET	StringRange
recenter	TEXT	NOPRESET	StringRange
drivePower	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
runControl	CHOICE		EnumRange (STANDBY OPERATE)
deice.state	TEXT	R/O	StringRange R/O
deice.heat	TEXT	R/O	StringRange R/O
deice.activate	TEXT	NOPRESET	StringRange
reset.az	TEXT	NOPRESET	StringRange
reset.el	TEXT	NOPRESET	StringRange
reset.tl	TEXT	NOPRESET	StringRange
reboot.az	TEXT	NOPRESET	StringRange
reboot.el	TEXT	NOPRESET	StringRange
reboot.tl	TEXT	NOPRESET	StringRange
tle.noradId	INTEGER		IntegerRange (0 .. 0)
tle.epochYear	INTEGER		IntegerRange (0 .. 0)
tle.epochDay	INTEGER		IntegerRange (0 .. 0)
tle.epochDayFrac	INTEGER		IntegerRange (0 .. 0)
tle.meanMotion1st	INTEGER		IntegerRange (0 .. 0)
tle.meanMotion2nd	INTEGER		IntegerRange (0 .. 0)
tle.dragTerm	INTEGER		IntegerRange (0 .. 0)
tle.inclination	INTEGER		IntegerRange (0 .. 0)
tle.rightAscension	INTEGER		IntegerRange (0 .. 0)
tle.eccentricity	INTEGER		IntegerRange (0 .. 0)
tle.argumentOfPerigee	INTEGER		IntegerRange (0 .. 0)
tle.meanAnomaly	INTEGER		IntegerRange (0 .. 0)
tle.meanMotion	INTEGER		IntegerRange (0 .. 0)
tle.set	TEXT	NOPRESET	StringRange
tle.offsetsEnable	CHOICE		EnumRange (DISABLED ENABLED)
tle.azOffset	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
tle.elOffset	FLOAT		DoubleRange (0.0000 .. 0.0000)
bcrx.port	INTEGER		IntegerRange (1 .. 8)
bcrx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
bcrx.loFrequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
bcrx.span	FLOAT		DoubleRange (0.0000 .. 0.0000)
bcrx.rbw	FLOAT		DoubleRange (0.000 .. 0.000)
bcrx.averages	INTEGER		IntegerRange (1 .. 255)
bcrx.measurementMode	CHOICE		EnumRange (DELTA PEAK)
bcrx.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
bcrx.frequencyDelta	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
bcrx.peakFrequency	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
steptrack.offsetTrack	CHOICE		EnumRange (DISABLED ENABLED)
steptrack.azOffset	FLOAT		DoubleRange (0.0000 .. 0.0000)
steptrack.elOffset	FLOAT		DoubleRange (0.0000 .. 0.0000)
steptrack.centerThresh	FLOAT		DoubleRange (0.00 .. 0.00)
steptrack.standbyDelay	INTEGER		IntegerRange (0 .. 0)
steptrack.lockThresh	FLOAT		DoubleRange (0.00 .. 0.00)
steptrack.lockState	TEXT	R/O	StringRange R/O
steptrack.peakState	TEXT	R/O	StringRange R/O
steptrack.waitingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
search.enable	CHOICE		EnumRange (DISABLED ENABLED)
search.beamwidth	FLOAT		DoubleRange (0.0000 .. 0.0000)
search.velocity	FLOAT		DoubleRange (0.0000 .. 0.0000)
search.stopLevel	FLOAT		DoubleRange (0.00 .. 0.00)
gsa.startTrack	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
gsa.stopTrack	TEXT	NOPRESET	StringRange
gsa.doPark	CHOICE	NOPRESET	EnumRange (NO YES)
gsa.azPark	FLOAT	NOPRESET	DoubleRange (0.0000 .. 0.0000)
gsa.elPark	FLOAT	NOPRESET	DoubleRange (0.0000 .. 0.0000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ prim. tripped)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ prim. inactive)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ prim. local)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ low softlimit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ high softlimit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ low hardlimit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ high hardlimit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ prim. e-stop)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ prim. jammed)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ prim. runaway)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AZ prim. break on)
faults.12	ALARM	R/O	AlarmFlagRange R/O (AZ prim. brake fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ prim. lash)
faults.14	ALARM	R/O	AlarmFlagRange R/O (AZ prim. sync)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (AZ prim. online)
faults.21	ALARM	R/O	AlarmFlagRange R/O (AZ sec. fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (AZ sec. inactive)
faults.23	ALARM	R/O	AlarmFlagRange R/O (AZ sec. local)
faults.24	ALARM	R/O	AlarmFlagRange R/O (AZ sec. e-stop)
faults.25	ALARM	R/O	AlarmFlagRange R/O (AZ sec. jammed)
faults.26	ALARM	R/O	AlarmFlagRange R/O (AZ sec. brake on)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AZ sec. brake fault)
faults.28	ALARM	R/O	AlarmFlagRange R/O (AZ sec. sync)
faults.29	ALARM	R/O	AlarmFlagRange R/O (AZ sec. online)
faults.30	ALARM	R/O	AlarmFlagRange R/O (AZ near s/w limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (EL prim. tripped)
faults.32	ALARM	R/O	AlarmFlagRange R/O (EL prim. inactive)
faults.33	ALARM	R/O	AlarmFlagRange R/O (EL prim. local)
faults.34	ALARM	R/O	AlarmFlagRange R/O (EL low softlimit)
faults.35	ALARM	R/O	AlarmFlagRange R/O (EL high softlimit)
faults.36	ALARM	R/O	AlarmFlagRange R/O (EL low hardlimit)

name	*type*	*flags*	*range*
faults.37	ALARM	R/O	AlarmFlagRange R/O (EL high hardlimit)
faults.38	ALARM	R/O	AlarmFlagRange R/O (EL prim. e-stop)
faults.39	ALARM	R/O	AlarmFlagRange R/O (EL prim. jammed)
faults.40	ALARM	R/O	AlarmFlagRange R/O (EL prim. runaway)
faults.41	ALARM	R/O	AlarmFlagRange R/O (EL prim. brake on)
faults.42	ALARM	R/O	AlarmFlagRange R/O (EL prim. brake fault)
faults.43	ALARM	R/O	AlarmFlagRange R/O (EL prim. lash)
faults.44	ALARM	R/O	AlarmFlagRange R/O (EL prim. sync)
faults.45	ALARM	R/O	AlarmFlagRange R/O (EL prim. online)
faults.51	ALARM	R/O	AlarmFlagRange R/O (EL sec. fault)
faults.52	ALARM	R/O	AlarmFlagRange R/O (EL sec. inactive)
faults.53	ALARM	R/O	AlarmFlagRange R/O (EL sec. local)
faults.54	ALARM	R/O	AlarmFlagRange R/O (EL sec. e-stop)
faults.55	ALARM	R/O	AlarmFlagRange R/O (EL sec. jammed)
faults.56	ALARM	R/O	AlarmFlagRange R/O (EL sec. brake on)
faults.57	ALARM	R/O	AlarmFlagRange R/O (EL sec. brake fault)
faults.58	ALARM	R/O	AlarmFlagRange R/O (EL sec. sync)

name	*type*	*flags*	*range*
faults.59	ALARM	R/O	AlarmFlagRange R/O (EL sec. online)
faults.61	ALARM	R/O	AlarmFlagRange R/O (TL prim. tripped)
faults.62	ALARM	R/O	AlarmFlagRange R/O (TL prim. inactive)
faults.63	ALARM	R/O	AlarmFlagRange R/O (TL prim. local)
faults.64	ALARM	R/O	AlarmFlagRange R/O (TL low softlimit)
faults.65	ALARM	R/O	AlarmFlagRange R/O (TL high softlimit)
faults.66	ALARM	R/O	AlarmFlagRange R/O (TL low hardlimit)
faults.67	ALARM	R/O	AlarmFlagRange R/O (TL high hardlimit)
faults.68	ALARM	R/O	AlarmFlagRange R/O (TL prim. e-stop)
faults.69	ALARM	R/O	AlarmFlagRange R/O (TL prim. jammed)
faults.70	ALARM	R/O	AlarmFlagRange R/O (TL prim. runaway)
faults.71	ALARM	R/O	AlarmFlagRange R/O (TL prim. brake on)
faults.72	ALARM	R/O	AlarmFlagRange R/O (TL prim. brake fault)
faults.73	ALARM	R/O	AlarmFlagRange R/O (TL prim. lash)
faults.74	ALARM	R/O	AlarmFlagRange R/O (TL prim. sync)
faults.75	ALARM	R/O	AlarmFlagRange R/O (TL prim. online)
faults.81	ALARM	R/O	AlarmFlagRange R/O (TL sec. fault)

name	*type*	*flags*	*range*
faults.82	ALARM	R/O	AlarmFlagRange R/O (TL sec. inactive)
faults.83	ALARM	R/O	AlarmFlagRange R/O (TL sec. local)
faults.84	ALARM	R/O	AlarmFlagRange R/O (TL sec. e-stop)
faults.85	ALARM	R/O	AlarmFlagRange R/O (TL sec. jammed)
faults.86	ALARM	R/O	AlarmFlagRange R/O (TL sec. brake on)
faults.87	ALARM	R/O	AlarmFlagRange R/O (TL sec. brake fault)
faults.88	ALARM	R/O	AlarmFlagRange R/O (TL sec. sync)
faults.89	ALARM	R/O	AlarmFlagRange R/O (TL sec. online)
faults.91	ALARM	R/O	AlarmFlagRange R/O (LNB 1)
faults.92	ALARM	R/O	AlarmFlagRange R/O (LNB 2)
faults.93	ALARM	R/O	AlarmFlagRange R/O (BDC 1)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Clock init)
faults.95	ALARM	R/O	AlarmFlagRange R/O (SNMP sync)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Decimator)
faults.97	ALARM	R/O	AlarmFlagRange R/O (De-ice power)

9.100 CandM-Switch

This page describes the device driver and the device window for the C&M Systems Microwave Waveguide Switching and HPA Protection system tested with **Model CM1208** with up to 2 waveguide switches and 2+1 HPA redundancy. The command set is similar to CPI Waveguide Switching Systems but additionally supports HPA Priority Setting.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

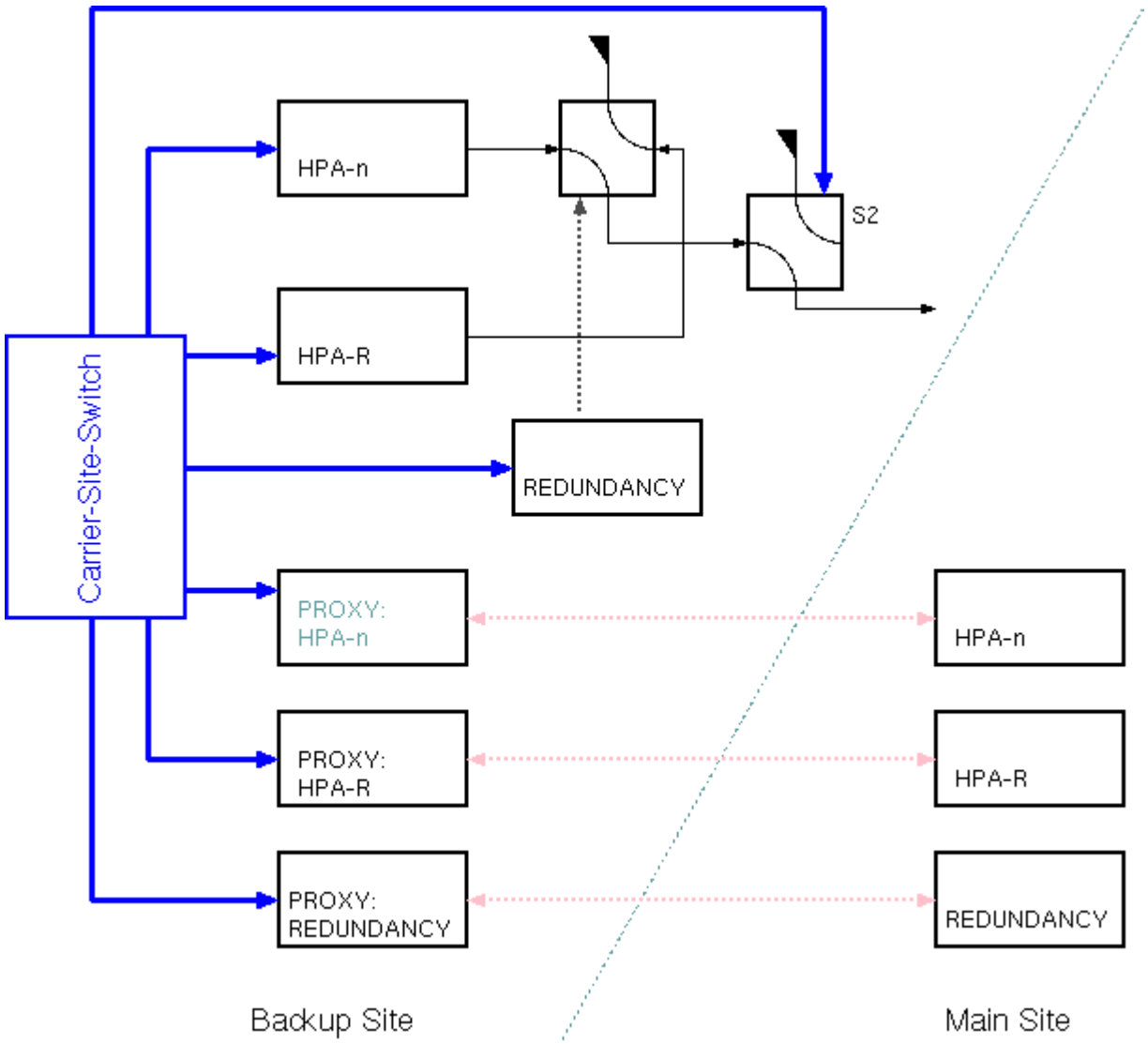
name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
priority	CHOICE		EnumRange (N/A HPA-01 HPA-02 HPA-03 HPA-04 HPA-05 HPA-06)
switch.01	CHOICE		EnumRange DIS (A B FLT)
switch.02	CHOICE		EnumRange DIS (A B FLT)
switch.03	CHOICE		EnumRange DIS (A B FLT)
switch.04	CHOICE		EnumRange DIS (A B FLT)
switch.05	CHOICE		EnumRange DIS (A B FLT)
switch.06	CHOICE		EnumRange DIS (A B FLT)
switch.07	CHOICE		EnumRange DIS (A B FLT)
switch.08	CHOICE		EnumRange DIS (A B FLT)
switch.09	CHOICE		EnumRange DIS (A B FLT)
switch.10	CHOICE		EnumRange DIS (A B FLT)
switch.11	CHOICE		EnumRange DIS (A B FLT)
switch.12	CHOICE		EnumRange DIS (A B FLT)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ext. interlock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unknown model)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HPA 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (HPA 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (HPA 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (HPA 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (HPA 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (HPA 6)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch 8)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch 9)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch 12)

9.101 Carrier-Site-Switch

The Carrier-Site-Switch logical device controls the equipment of one transmit chain for a site diversity application between two uplink ground stations.



As shown in the figure above, the Carrier-Site-Switch logical device is operated at the Backup site. There it controls the HPA which shall take over the Main uplink, the redundant HPA, the protection switch logical device and the waveguide switch which routes the HPA output either to dummy load or to the antenna. Beside this, the Carrier-Site-Switch controls three Proxy devices which must be configured to resemble the corresponding HPAs and the protection switch at the main site.

Site switch procedure

Switching the uplink from the main site to backup is a 2-step procedure. With the first step, the backup site gets prepared to take over the uplink, the real site switch then is performed as the second step. The description below uses a configuration with HPAs controlled by a N:1 redundancy at both sites. This is only an example, the Carrier-Site-Switch device may be configured to work with various transmit devices with or without redundancy. This may be separately configured for both sites.

To prepare the backup site to take over the uplink, the function 'prepare' performs the following

operations:

1. Switch off the the protection for the HPA of interest at the protection switch if a N-TO-1 protection is configured
2. Detect if HPA-? or the redundant one must be used or both in case of a 1-TO-1 redundancy
3. Switch the HPA(s) of interest to RF-OFF.
4. After waiting the bckpTxChgReadDelay poll the HPA up to 10 seconds until it states RF-OFF. Abort on failure.
5. Switch the HPA BEAM-ON
6. Switch the transmit signal to antenna using the dummyLoadSwitch.

The backup transmit facility is signalled to be prepared, when the Carrier-Site-Switch detects the protection switch to be OFF, the HPA to be RF-OFF / BEAM-ON and S2 set to antenna.

The actual switch over can only be performed, if the 'prepared' condition is detected by the Carrier-Site-Switch. This enables the 'BACKUP' button at the user interface. The steps of the switch over are:

1. Switch OFF protection for the main site HPA.
2. Detect if HPA-? or the redundant one must be used at the main site or both in case of a 1-TO-1 protection.
3. Switch the main site HPA(s) to RF-OFF.
4. After waiting the mainTxChgReadDelay poll the HPA up to 10 seconds until it states RF-OFF. Abort on failure.
5. Switch the backup site HPA(s) to RF-ON.
6. After waiting the bckpTxChgReadDelay poll the HPA up to 10 seconds until it states RF-ON. Abort on failure.

Switching back to the main site works the same way the other way round.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands actual uplink location.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the devices that will perform the carrier on/off settings.

parameter	***description***

parameter	***description***
dummyLoadSwitch	The device name of the waveguide switch which routes the backup site uplink either to the antenna or to a dummy load.
dlSwitchPositions	Two switch positions for the this switch, separated by a comma. The first value is used to switch to dummy load, the second one to switch it to antenna.
bckpProtectMode	Selects the type of protection switch used with the TX equipment at the backup site. May be one of NONE, 1-TO-1 or N-TO-1, should be set to the value describing the protection mode of the TX equipment.
bckpProtectDevice	The name of the N:1 protection controller device used at the backup site. Actually the Carrier-Site-Switch supports the Protection-Switch-N-To-1 logical device for this and devices which resemble the control parameter structure if this device. Must be set if the bckpProtectMode is N-TO-1, for other protection mode settings this parameter is not required.
bckpTxDevice	The device name of the transmit device (typically an HPA) which produces the carrier at the backup site. Any device processing the carrier and providing a 'tx.on' parameter to switch the carrier on/off may be configured for this.
bckpTxRedDevice	The name of the redundant transmit device in cases where a bckpProtect mode is not NONE. For a symmetrical 1:1 redundancy this is the name of the second HPA (see redundancy configuration below.
bckpChainNo	If the backup site is configured for a N-TO-1 protection, this must be set to the chain number of the bckpTxDevice in the protection device (1..12).
mainProtectMode	Selects the type of protection switch used with the TX equipment at the main site. May be one of NONE, 1-TO-1 or N-TO-1, should be set to the value describing the protection mode of the TX equipment.
mainProtectDevice	The name of the N:1 protection controller device used at the main site. Actually the Carrier-Site-Switch supports the Protection-Switch-N-To-1 logical device for this and devices which resemble the control parameter structure if this device. Must be set if the mainProtectMode is N-TO-1, for other protection mode settings this parameter is not required.

parameter	***description***
mainTxDevice	The device name of the transmit device (typically an HPA) which produces the carrier at the main site. Any device processing the carrier and providing a 'tx.on' parameter to switch the carrier on/off may be configured for this.
mainTxRedDevice	The name of the redundant transmit device in cases where a mainProtect mode is not NONE. For a symmetrical 1:1 redundancy this is the name of the second HPA (see redundancy configuration below.
mainChainNo	If the main site is configured for a N-TO-1 protection, this must be set to the chain number of the mainTxDevice in the protection device (1..12).
mainName	A short (one word) name for the main site
mainDescription	A descriptive name for the main site. This appears in event log entries and in status messages of the device, hence it should not be to long (max 15 characters are recommended)
bckpName	A short (one word) name for the backup site
bkcpDescription	A descriptive name for the backup site. This appears in event log entries and in status messages of the device, hence it should not be to long (max 15 characters are recommended)
comment	The text is presented in the device's GUI as a description of the carrier which is switched between the sites.
bckpTxChgReadDelay	This parameters sets the delay between an RF ON/OFF command sent to the backup TX device and the first read back which is done to check if the carrier is really at the commanded state. Some devices echo the RF ON/OFF state without switching the carrier and report the true state two seconds later because the command could not be executed as expected. Increasing the delay can help to suppress the consequences of this effect.
mainTxChgReadDelay	This parameters sets the delay between an RF ON/OFF command sent to the main TX device and the first read back which is done to check if the carrier is really at the commanded state. Some devices echo the RF ON/OFF state without switching the carrier and report the true state two seconds later because the command could not be executed as expected. Increasing the delay can help to suppress the consequences of this effect.

parameter	*description*
abortOn	The CarrierSiteSwitch aborts a preparation or switchover procedure if one of the transmit devices reports a fault. This setting selects if this shall happen in case of a FAULT or also if the device reports a WARNING.

Redundancy configuration

The Carrier-Site-Switch supports three different modes of redundancy for the transmit devices. These modes are configured individually for both sites, this means you can have two 1:1 redundante HPAs at the one site and at the other site HPAs which are part of a N:1 redundancy. The three redundancy modes are:

- **NONE:** There is no redundancy at all for this site. The device only controls the primary TX device at this site and skips all steps regarding the redundancy.
- **1-TO-1:** The TX devices at this site are in a symmetric 1:1 redundancy. The device always controls both devices at the same time, their carriers are simultaneously switched on/off, the carrier is assumed to be off if both devices report RF OFF. The device skips all the steps described above regarding the redundancy.
- **N-TO-1:** The TX devices at this site are in a N:1 redundancy. The device behaves as described above, it uses the redundant device if the protection controller uses this in place of the configured primary device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mainProtectMode	TEXT	R/O	StringRange R/O
info.bckpProtectMode	TEXT	R/O	StringRange R/O
info.mainHasBeamOn	TEXT	R/O	StringRange R/O
info.bckpHasBeamOn	TEXT	R/O	StringRange R/O
info.hasDummyLoadSwitch	TEXT	R/O	StringRange R/O
config.dummyLoadSwitch	TEXT	SAVE SETUP	StringRange
config.dlSwitchPositions	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.bckpProtectMode	CHOICE	SAVE SETUP	EnumRange (NONE 1-TO-1 N-TO-1)
config.bckpProtectDevice	TEXT	SAVE SETUP	StringRange
config.bckpTxDevice	TEXT	SAVE SETUP	StringRange
config.bckpTxRedDevice	TEXT	SAVE SETUP	StringRange
config.bckpTxChainNo	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.mainProtectMode	CHOICE	SAVE SETUP	EnumRange (NONE 1-TO-1 N-TO-1)
config.mainProtectDevice	TEXT	SAVE SETUP	StringRange
config.mainTxDevice	TEXT	SAVE SETUP	StringRange
config.mainTxRedDevice	TEXT	SAVE SETUP	StringRange
config.mainTxChainNo	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.mainName	TEXT	SAVE SETUP	StringRange
config.mainDescription	TEXT	SAVE SETUP	StringRange
config.bckpName	TEXT	SAVE SETUP	StringRange
config.bckpDescription	TEXT	SAVE SETUP	StringRange
config.comment	TEXT	SAVE SETUP	StringRange
config.bckpTxChgReadDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 15000)

name	*type*	*flags*	*range*
config.mainTxChgReadDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 15000)
config.abortOn	CHOICE	SAVE SETUP	EnumRange (WARNING FAULT)
config.autoPrepare	CHOICE	SAVE SETUP	EnumRange (OFF ON)
main	TEXT		StringRange
bckp	TEXT		StringRange
prepare	TEXT		StringRange
release	TEXT		StringRange
progressReport	TEXT	R/O	StringRange R/O
progressIcon	TEXT	R/O	StringRange R/O
reset	TEXT		StringRange
prepared	BOOLEAN	R/O	BooleanRange R/O (true false)
location	TEXT	R/O	StringRange R/O
save.state	INTEGER	SAVE	IntegerRange (0 .. 0)
save.mainProtection	TEXT	SAVE	StringRange

9.102 Channel-Select

There is no special online help for this type of device. Please refer to the general description of device windows instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.satposId	TEXT	SAVE SETUP	StringRange
config.rxtx	CHOICE	SAVE SETUP	EnumRange (RX TX)
config.polMode	CHOICE	SAVE SETUP	EnumRange (XY HV AB BA)
channel	CHOICE		EnumRange (?)
show	CHOICE	SAVE	EnumRange (ALL SAT)

9.103 ChinaSuncom-9293S3-ACU

This page describes the device driver and the device window for the China Suncom 9293S3 Antenna Controller.

The driver controls the ACU via Modbus/TCP, in the VLC Setup a ChinaSuncom-9293S3-ACU device must be attached to an interface / ports with the ACU's IP address and port 502.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing. The page contains all parameters of te ACU managed by the driver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set for a proper operation of the ACU.

parameter	*description*
address	Doesn't care, may be left empty. Modbus/TCP doesn't use device addresses.

parameter	*description*
azLoLimit azHiLimit elLoLimit elHiLimit plLoLimit plHiLimit	Sets the valid range of the three pointing angles. You should set these to the values of the soft limit as set in the ACU itself. Unfortunately there is no way to read these values via Modbus.

Remarks

The operation of the Suncom ACU is based on a number operation modes. You first have to select the operation mode, then you can enter parameters or command specific to this mode.

- To move the antenna to given pointing angles, set 'operation mode' to 'PRESET' and then enter the target angles.
- To preset the antenna for a given satellite number, set 'operation mode' to 'SATPRESET' then enter the satellite number and click 'GO'
- To start steptrack, set 'operation mode' to 'STEPTRACK' and then select the steptrack mode (OFF/ON).
- To stop the antenna, click 'STOP'
- 'operation mode' may be set to 'MANUAL', but the driver offers no buttons to jog the antenna. The response would be too slow via Modbus for a satisfying user experience.
- For machine controlled operation, pointing angles may be commanded in any operation mode. The driver automatically switched the ACU to 'PRESET' mode if it receives a pointing angle command.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
internal.satno	INTEGER	R/O	IntegerRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.azUp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.azDn	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.elUp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.elDn	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.p1Up	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.moving.p1Dn	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.p2Up	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.p2Dn	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
function	CHOICE		EnumRange (IDLE PRESET MANUAL STEPTRACK SATPRESET)
tracking	CHOICE		EnumRange (OFF ON)
preset	CHOICE		EnumRange (OFF ON)
satpreset	CHOICE		EnumRange (OFF ON)
satno	INTEGER		IntegerRange (1 .. 100)
stop	TEXT	NOPRESET	StringRange
internal.posCommanded	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.fncCommanded	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limig)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (P1 high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (P1 low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (P2 high limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (P2 low limit)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ angle sensor)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL angle sensor)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL angle sensor)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Hand terminal control)
faults.13	ALARM	R/O	AlarmFlagRange R/O (GPS status)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (IO expander)

9.104 ChinaSuncom-Beacon-Receiver-SC9800

This page describes the device driver and the device window for the China Suncom Beacon Receiver SC9800 via SNMP.

It supports beacon, dvb and detection mode and was tested with

- SW version V30.3.4.2-830

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit as well as for initial setup.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write

community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- lnbVolt --- Sets the lnb voltage.
- lnb22KhzTone --- Sets the lnb tone/22kHz.
- agcRange --- Sets the agc voltage range: 3,3V 5V or 10V.
- beacon.mapMin --- Sets the mapped lower beacon level in dBm.
- beacon.mapMax --- Sets the mapped upper beacon level in dBm.
- dvb.mapMin --- Sets the mapped lower dvb level in dBm.
- dvb.mapMax --- Sets the mapped upper dvb level in dBm.
- detect.mapMin --- Sets the mapped lower detection level in dBm.
- detect.mapMax --- Sets the mapped upper detection level in dBm.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.ctrlSerial	TEXT	R/O	StringRange R/O
info.ctrlVersion	TEXT	R/O	StringRange R/O
info.ctrlHwVersion	TEXT	R/O	StringRange R/O
config.lnbVolt	CHOICE	SETUP	EnumRange (OFF 13.4V 18.2V 14.6V 18.2V)
config.22kHzTone	CHOICE	SETUP	EnumRange (OFF ON)
config.agcRange	CHOICE	SETUP	EnumRange (3.3V 5V 10V)
config.beacon.mapMin	INTEGER	SETUP	IntegerRange (-110 .. -20)
config.beacon.mapMax	INTEGER	SETUP	IntegerRange (-110 .. -20)
config.dvb.mapMin	INTEGER	SETUP	IntegerRange (-2 .. 23)
config.dvb.mapMax	INTEGER	SETUP	IntegerRange (-2 .. 23)
config.detect.mapMin	INTEGER	SETUP	IntegerRange (-110 .. -20)
config.detect.mapMax	INTEGER	SETUP	IntegerRange (-110 .. -20)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
opMode	CHOICE		EnumRange (DVBS2X DETECTION BEACON)
attenuation	INTEGER		IntegerRange (0 .. 30)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
dvb.refreshCycle	INTEGER		IntegerRange (4 .. 60)
dvb.frequency	FLOAT		DoubleRange (950.000 .. 2200.000)
dvb.symbolrate	FLOAT		DoubleRange (0.200 .. 500.000)
dvb.searchRange	INTEGER		IntegerRange (500 .. 70000)

name	*type*	*flags*	*range*
dvb.rolloff	CHOICE		EnumRange (35% 25% 20% 10% 5% 2% UNKNOWN)
detect.refreshCycle	INTEGER		IntegerRange (6 .. 60)
detect.frequency	FLOAT		DoubleRange (950.000 .. 2200.000)
detect.symbolrate	FLOAT		DoubleRange (0.200 .. 500.000)
detect.searchRange	INTEGER		IntegerRange (100 .. 5000)
detect.rolloff	INTEGER		IntegerRange (1 .. 35)
beacon.refreshCycle	INTEGER		IntegerRange (6 .. 60)
beacon.frequency	FLOAT		DoubleRange (950.000 .. 2200.000)
beacon.aquisitionRange	INTEGER		IntegerRange (50 .. 245)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Receiver Lock)

9.105 Choice-Translator

The Choice-Translator logical device provides a 2-way translation table for choice parameters. A common application for the Choice-Translator is to rename switch positions from the predefined ON/OFF resp. A/B labels to application specific designations like 'Transmit path 1'.

To make use of the Choice-Translator, install it in the device setup, configure the translation tables and the message ID you want to translate. In your user screens operate the translated parameter now through the 'value' parameter of the Choice-Translator instead of directly via the parameters original message ID.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Simply contains a choice box to operate the translated parameter. Provided for completeness and for testing purposes.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Choice-Translator device with the device / parameter it shall translate.

parameter	***description***
messageId	Enter the message identifier here of the parameter which shall be translated, eg. "SWITCH-1. position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
choices	Enter the *original* choices of the parameter to translate, separated by commas. Ensure that all choices defined for this parameter are contained and that they are written exactly the same than the original. There must be no additional spaces between the items.
newChoices	Enter the translated list of choices here. There must be the same number of items in 'choices' and 'newChoices'.
fileName	The name of a file to read the 'choices' / 'newChoices' from. See **Translation File** below for details.

Translation File

Instead of defining the 'choices' / 'newChoices' lists explicitly, you may read these settings from a file. This gives the opportunity to define recurring translation settings once and re-use them in multiple Choice-Translator instances. The file must define one pair of choice / newChoice value per line, separated by a comma character. Empty lines and lines beginning with a '#' character are ignored. Example:

```
# Example Choice-Translator definition file
#
OFF,NO
ON,YES
```

This translates the device settings OFF/ON to the values NO/YES in the Choice-Translator. To use this feature, edit a file with the translation definition and enter the name of the file in the 'fileName' configuration setting of the Choice-Translator. This sets the 'choices' / 'newChoices' values to the lists parsed from the file and makes these settings read-only. To return to normal mode, enter an empty String to the 'fileName' parameter.

Remarks

If the controlled parameter changes to a value which is not found in 'choices', the translated value does not change. The other way round, if 'choices' contains a value not recognized by the controlled parameter, the latter will change in a non predictable way.

Be aware, that there are some sophisticated device drivers which dynamically change the available choices (e.g. a modem may allow certain FEC settings only on QPSK, not in BPSK mode). Controlling such parameters with a Choice-Translator may end up in confusing results.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.messageId	TEXT	SAVE SETUP	StringRange
config.choices	TEXT	SAVE SETUP	StringRange
config.newChoices	TEXT	SAVE SETUP	StringRange
config.fileName	TEXT	SAVE SETUP	StringRange
value	CHOICE		EnumRange ()
reset	TEXT	NOPRESET	StringRange

9.106 Choice

The logical device Choice provides a dropdown field which could be used to shows and saves a selection. By default "ON" and "OFF" are set as possible options, but this could be changed in the device setup. The choice field is also lockable by pressing a button which means that you can set it to readonly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the choice and lock button
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	*description*
choice	A comma separated list with all choices which should be available in the dropdown list
label	Sets the label of the comment field in the device window. (not required)

Choice and lock status are saved on the M&C system and preserved after a system restart

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.choices	TEXT	SAVE SETUP	StringRange
config.label	TEXT	SAVE SETUP	StringRange
comment	CHOICE	SAVE	EnumRange (OFF ON)
lock	BOOLEAN	SAVE	BooleanRange (true false)

9.107 Cibred-LAB4

This page describes the readonly device driver and the device window for Cibred Sud LAB4 waveguide pressurizer/ dehydrator via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device status parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.runTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
pressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low Pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High Pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High Humidity)

9.108 Cisco-2800GSM-SNMP

This page describes the device driver and the device window for Cisco 2800 router with

3G/GSM card. It monitors only the 3G/GSM specific parameters. It is possible to combine an instance of this driver (device) with other MIB2 or Router drivers on the same interface with protocol type SNMP to get additional parameters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the Main Receive Parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

These are the driver configuration parameters of the Cisco-Router driver:

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- radioInterfaceIdentifier --- interface number of 3G/GSM card
- rssiLowThreshhold --- alarm threshold for 3G/GSM signal

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
radioInterfaceIdentifier	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
rssiLowThreshhold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
c3gGsmCountry	TEXT	R/O	StringRange R/O
c3gGsmNetwork	TEXT	R/O	StringRange R/O
c3gGsmPacketService	TEXT	R/O	StringRange R/O
c3gCurrentGsmRssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
c3gGsmCurrentBand	TEXT	R/O	StringRange R/O
rttMonLatestRttOperSense	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Signal Strengh)
faults.02	ALARM	R/O	AlarmFlagRange R/O (VPN-Connection)

9.109 Cisco-2900-SNMP

This page describes the device driver and the device window for Cisco 2900 router with 3G/GSM card. It monitors only the 3G/GSM specific parameters and OSPF routing protocol information. The drivers detects which on of the configured routes is active (up to 8) and therefore it show which way the data takes at the moment. For example via 3G mobile network or via satellite link.

It is possible to combine an instance of this driver (device) with other MIB2 or Router drivers on the same interface with protocol type SNMP to. get additonal parameters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the Main Receive Parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

These are the driver configuration parameters of the Cisco-Router driver:

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- radioInterfaceIdentifier --- interface number of 3G/GSM card
- rssiLowThreshhold --- alarm threshold for 3G/GSM signal
- config.routeOID.01-08 --- OID where the driver looks for an active route
- config.routeIf.01-08 --- interface number which indicates that a route is active

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
radioInterfaceIdentifier	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
rssiLowThreshhold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeOID.01	TEXT	SAVE SETUP	StringRange
config.routeOID.02	TEXT	SAVE SETUP	StringRange
config.routeOID.03	TEXT	SAVE SETUP	StringRange
config.routeOID.04	TEXT	SAVE SETUP	StringRange
config.routeOID.05	TEXT	SAVE SETUP	StringRange
config.routeOID.06	TEXT	SAVE SETUP	StringRange
config.routeOID.07	TEXT	SAVE SETUP	StringRange
config.routeOID.08	TEXT	SAVE SETUP	StringRange
config.routeIf.01	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.02	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.routeIf.03	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.04	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.05	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.06	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.07	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.routeIf.08	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
internal.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
c3gGsmCountry	TEXT	R/O	StringRange R/O
c3gGsmNetwork	TEXT	R/O	StringRange R/O
c3gCurrentGsmRssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
c3gGsmCurrentBand	TEXT	R/O	StringRange R/O
routeActive.01	BOOLEAN		BooleanRange (true false)
routeActive.02	BOOLEAN		BooleanRange (true false)
routeActive.03	BOOLEAN		BooleanRange (true false)
routeActive.04	BOOLEAN		BooleanRange (true false)
routeActive.05	BOOLEAN		BooleanRange (true false)

name	*type*	*flags*	*range*
routeActive.06	BOOLEAN		BooleanRange (true false)
routeActive.07	BOOLEAN		BooleanRange (true false)
routeActive.08	BOOLEAN		BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Signal strength)
faults.02	ALARM	R/O	AlarmFlagRange R/O (VPN connection)

9.110 Cisco-D9800-IRD

This page describes the device driver and the device window for a Cisco/Synamedia IRDs D9800 series. It supports up to 4 audio channels. **Note:** As the unit does not allow to change LO or receive frequency out of the rf range in one snmp command the usage of the device stored lo frequency can be switched OFF using config.useLoFreq. In this case the device stored LO Frequency will not be changed at any time by the driver and is used only for calculation. This mode is useful when operating the unit together with a switch matrix, a CDB and/or RX-Channel device.

It is tested with the following models:

- D9854_4x0 SW: 4.8.0
- D9800-SS-MPEGOIP SW: 4.8.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The Rf monitoring and control parameters.
- --- Contains IP in parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHz Auto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power
- lnbVoltage --- Selects the voltage level applied to the input if lnbPower is enabled.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter below becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- deviceComment --- An optional text entered here appears as an addition to the tooltip

which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

- audioChannels --- Defines the number of audio channels (2 or 4) of the unit.
- useLoFreq --- As the unit does not allow to change lo or receive frequency out of the given rf range in one snmp command the usage of the device stored lo frequency can be switched off using config.useLoFreq.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.trackingId	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.bandEdge	INTEGER	SETUP	IntegerRange (0 .. 0)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.audioChannels	CHOICE	SAVE SETUP	EnumRange (2 4)
config.useLoFreq	CHOICE	SAVE SETUP	EnumRange (YES NO)
idx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.band	TEXT	R/O	StringRange R/O
internal.frequency	FLOAT		DoubleRange (0.0 .. 0.0)
input	CHOICE		EnumRange (1 2 3 4)
interface	CHOICE		EnumRange (ASI SAT IP NONE)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	CHOICE		EnumRange (X Y)
dvbmode	CHOICE		EnumRange (DVB-S DVB-S2)
modulation	CHOICE	R/O	EnumRange R/O (QPSK 8PSK 16QAM)
fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8 AUTO)
rolloff	CHOICE		EnumRange (35% 25% 20% AUTO)
pilots	CHOICE	R/O	EnumRange R/O (OFF ON N/A)
symbolRate	FLOAT		DoubleRange (1.000 .. 45.000)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
programNo	INTEGER		IntegerRange (0 .. 0)
programList	CHOICE		EnumRange (0 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ber	TEXT	R/O	StringRange R/O
state.fec	TEXT	R/O	StringRange R/O
video.profile	TEXT	R/O	StringRange R/O
video.format	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.aspect	CHOICE		EnumRange (4:3 16:9)
video.arConvert	CHOICE		EnumRange (NONE AUTO AUTO-AFD 16:9-LETTERBOX 4:3-PILLARBOX 14:9 4:3-CCO 16:9-SCALE)
video.primary	CHOICE		EnumRange (AUTO 1080i 720p SD)
video.sd	CHOICE		EnumRange (AUTO NTSC PAL-BG PAL-D PAL-I PAL-M PAL-NAR NTSC-J)
video.sync	CHOICE		EnumRange (ON OFF)
subtitle.mode	CHOICE		EnumRange (OFF ON IMITEXT DVB)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.language	TEXT	R/O	StringRange R/O
audio.1.mode	CHOICE		EnumRange (STEREO MIXED LEFT-MONO RIGHT-MONO)
audio.1.dd	CHOICE		EnumRange (PCM COMPRESSED)
audio.1.ddPlus	CHOICE		EnumRange (PASSTHROUGH TRANSCODING)

name	*type*	*flags*	*range*
audio.1.pmtSource	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.language	TEXT	R/O	StringRange R/O
audio.2.mode	CHOICE		EnumRange (STEREO MIXED LEFT-MONO RIGHT-MONO)
audio.2.dd	CHOICE		EnumRange (PCM COMPRESSED)
audio.2.ddPlus	CHOICE		EnumRange (PASSTHROUGH TRANSCODING)
audio.2.pmtSource	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.language	TEXT	R/O	StringRange R/O
audio.3.mode	CHOICE		EnumRange (STEREO MIXED LEFT-MONO RIGHT-MONO)
audio.3.dd	CHOICE		EnumRange (PCM COMPRESSED)
audio.3.ddPlus	CHOICE		EnumRange (PASSTHROUGH TRANSCODING)

name	*type*	*flags*	*range*
audio.3.pmtSource	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.language	TEXT	R/O	StringRange R/O
audio.4.mode	CHOICE		EnumRange (STEREO MIXED LEFT-MONO RIGHT-MONO)
audio.4.dd	CHOICE		EnumRange (PCM COMPRESSED)
audio.4.ddPlus	CHOICE		EnumRange (PASSTHROUGH TRANSCODING)
audio.4.pmtSource	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
ca.bissmode	CHOICE		EnumRange (BISS-1 BISS-E)
ca.bisskey	TEXT		StringRange
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ip	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.interface	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (1 .. 65534)
ipin.fecMode	CHOICE		EnumRange (OFF 1D 2D)
ipin.fecColumnsPort	INTEGER		IntegerRange (2 .. 65534)
ipin.fecRowsPort	INTEGER		IntegerRange (2 .. 65534)
ipin.dejitter	CHOICE		EnumRange (VBR CBR)
ipin.bufferDuration	INTEGER		IntegerRange (0 .. 2147483647)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Audio 4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Warning)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.111 Cisco-Monitor

The *Cisco-Monitor* device driver monitors a small set of parameters at a Cisco Router by means of SNMP. The monitored parameters are:

- The processor load, averaged over 5 minutes
- The size of largest available memory block
- The operation state of the primary power supply
- The operation state of the redundant power supply
- The standby state of up to 8 HSRP groups The driver permits to configure the list of HSRP group stated to be monitored. For this, the table index of each group to monitor must be entered at the maintenance/setup page of the device window.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational state of each interface monitored.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the device in the sat-nms software.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- hsrpIndex.1 ... hsrpIndex.8 --- For each group to be monitored, the table index of the group has to be entered here. These indexes have the form "x.y" where x and y are two integer numbers. Example values are "7.11", "8.12", "9.13" etc. The best way to find out the table indexes configured for this individual device is to perform a "walk" through the HSRP group table (OID 1.3.6.1.4.1.9.9.106.1.2.1.1.15) of the device by means of a SNMP MIB browser tool. Setting an index value to an empty string means not to monitor this value.
- config.hsrpDesc.1 ... config.hsrpDesc.8 --- Description for each HSRP group

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysDescr	TEXT	R/O	StringRange R/O
config.hsrpIndex.1	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.2	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.3	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.4	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.5	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.6	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.7	TEXT	SAVE SETUP	StringRange
config.hsrpIndex.8	TEXT	SAVE SETUP	StringRange
hsrpStandbyState.1	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.2	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.3	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.4	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.5	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
hsrpStandbyState.6	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.7	TEXT	R/O	StringRange R/O DIS
hsrpStandbyState.8	TEXT	R/O	StringRange R/O DIS
config.hsrpDesc.1	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.2	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.3	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.4	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.5	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.6	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.7	TEXT	SAVE SETUP	StringRange
config.hsrpDesc.8	TEXT	SAVE SETUP	StringRange
cpmCPUTotal5minRev	INTEGER	R/O	IntegerRange R/O (0 .. 100)
ciscoMemoryPoolLargestFree	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ciscoEnvMonSupplyState.1	TEXT	R/O	StringRange R/O
ciscoEnvMonSupplyState.2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PS 1 not present)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (PS 2 not present)

9.112 Cisco-Router

This page describes the device driver and the device window for Cisco routers. The driver works with Cisco 2500, 3640, 7513 routers. It provides some status/statistics polling via the console interface of the router. The driver reads the interface list and the BGP neighbor list at the router's console port, extracts some information from these lists and shows them to the operator as formatted tables. Polling is repeated as fast as the router replies.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the list of interfaces installed at the router. For each interface it's state and some statistics is displayed. The device driver resets the error counters every 10 minutes to make the numbers comparable.
- --- Displays the BGP neighbour list for the router.
- --- This window provides an enlarged version of the low level command interface. The command/reply fields are the same as at the maintenance page, however the enlarged size makes it more comfortable to operate the router at it's console command line.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

These are the driver configuration parameters of the Cisco-Router driver:

parameter	*description*
address	There is no device address used with this device driver. Leave this field empty.
crcThreshold	The threshold value for CRC faults. The driver raises a 'General' fault if at least one interface reports more CRC faults in 10 minutes than this value.
ifcResetThreshold	The threshold value for interface resets. The driver raises a 'General' fault if at least one interface reports more interfaces resets in 10 minutes than this value.

parameter	*description*
carrierTransThreshold	The threshold value for carrier transitions. The driver raises a 'General' fault if at least one interface reports more carrier transitions in 10 minutes than this value.
Serial	Remember to set the line parameters for the serial interface connected to the router. The router expects a line setting of *9600,N,8,1* .

Hint for scrolling displays

The interface list and BGP list displays have been reported to appear with missing or wrong scaled scroll bars on some computers. Resizing the device window a small amount (drag the lower right corner with the mouse) remedies this flaw.

Event logging

Although the driver provides only one common fault flag for all interfaces, it monitors the up/down states and the threshold comparisons for each interface in particular. If the driver recognizes a change of one of these states, it sends a message to the event log. By this messages, you may evaluate at which time an interface of the router was OK or not. As this information is based on cyclic polling, it may happen that short time drop outs are not listed in the log.

Fault recognition

The driver summarizes the monitored interface states and generates one common fault flag from this. The 'General' fault flag is set if at least one interface

- is administrative 'UP' but operational 'DOWN' or
- shows more CRC faults than the threshold value set on the maintenance page or
- shows more interface resets than the threshold value set on the maintenance page or
- shows more carrier transitions than the threshold value set on the maintenance page.

Low level commanding

The router device driver provides an extended low level command interface. At the page you may enter commands for testing (e.g. ping or traceroute) or you may log in to change the router's configuration at the console prompt. When using this feature, you should consider the following:

- Commands are inserted into the normal polling cycle of the driver. Depending on the number of interfaces installed in the router, if may take a couple of seconds until the driver will have finished it's polling cycle. When the driver has sent the command to the router, three dots are shown in the reply field.
- The driver sends everything coming back from the router to the reply field. With the standard 2000 msec timeout set at the serial interface, the display gets updated every 2 seconds. You can watch the progress of a ping or traceroute command this way. The

driver returns to it's normal operation after it recognizes the router name (the prompt) at the end of the reply.
- Never try to change the router's name with a low level command. The driver will not find the end of the reply to this command and will remain in this state until you restart the VLC / M&C system.
- You may interrupt a traceroute command by sending a "" (backslash) command while the traceroute is running.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
config.crcThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifcResetThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.carrierTransThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
interfaces	TEXT	R/O	StringRange R/O
bgp	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)

9.113 Codan-7586L

This page describes the device driver and the device window for the for the former Codan, now CPI redundancy controller 7586L via 7586L Controller using SNMP.

- Codan-BUC-7586L should be used for monitor and control of the BUCs/SSPAs itself.

It was tested with:

- 1:1 BUCs Type 4950L-W/E-AC/EX, no LNBs
- Firmware 90-20627-001

Default community string for this unit is rbuccom

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch parameters.
- --- Not used for this model. Available readings are placed on the info page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device. Default community string for this unit is rbuccom.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community. Default community string for this unit is rbuccom.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- switch --- Allows to set the installed switches for this controller: RX+TX,RX/TX,TX Only

- reference --- Allows to choose between using Internal or External reference source.
- referenceThresh --- Allows to set the reference level threshold in dBm.
- lnbThresh --- Allows to set the lnb current threshold in mA.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.reference	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.lnbVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.switch	CHOICE	SETUP	EnumRange (Rx+TX RX/TX TX Only)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)

name	*type*	*flags*	*range*
config.referenceThresh	FLOAT	SETUP	DoubleRange (-9.9 .. 4.9)
config.lnbThresh	INTEGER	SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (A B)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (DC 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (DC 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (BUC 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (BUC 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (TX WG Switch)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Ext Ref BUC)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.114 Codan-BUC-7586L

This page describes the device driver and the device window for the for the former Codan, now CPI blockupconverters/SSPAs via 7586L Controller using SNMP.

For selecting the respective BUC (A or B) the config.bucNumber has to be set in the device setup.

- Codan-7586L should be used for optional monitor and control of the redundancy unit itself.

It was tested with:

- 1:1 BUCs Type 4950L-W/E-AC/EX, no LNBs
- Firmware 90-20627-001

Default community string for this unit is rbuccom.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device. Default community string for this unit is rbuccom.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community. Default community string for this unit is rbuccom
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- txPowerTresh --- Allows to set the transmit power threshold in dBm.
- burstPowerTresh --- Allows to set the burst power threshold in dBm.
- txAtPowerUp --- Allows to choose the transmit startup behaviour (e.g after a reboot/power cycle): OFF or LAST STATE:
- redundancyMode --- Allows to choose the redundancy mode for the standby SSPA/BUC: NONE,WARM or HOT.
- reference --- Allows to choose between using Internal or External reference source.

- bucNumber --- Allows to choose the buc which should be controlled.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.loFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.minBurst	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
info.maxBurst	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.txPowerTresh	FLOAT	SETUP	DoubleRange (0.0 .. 50.0)
config.burstPowerTresh	FLOAT	SETUP	DoubleRange (0.0 .. 50.0)

name	*type*	*flags*	*range*
config.txAtPowerUp	CHOICE	SETUP	EnumRange (OFF LAST STATE)
config.redundancyMode	CHOICE	SETUP	EnumRange (NONE WARM HOT)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.bucNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.attn	INTEGER		IntegerRange (0 .. 20)
tx.frequency	INTEGER		IntegerRange (0 .. 0)
tx.ifFrequency	INTEGER		IntegerRange (0 .. 0)
location	CHOICE		EnumRange (ONLINE STANDBY)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.burstLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.reference	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.rfmin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.rfmax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.ifmin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.ifmax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lo1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lo2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.maxPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PA)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LO unlocked)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.07	ALARM	R/O	AlarmFlagRange R/O (LNB)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Redundancy Cont.)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.115 Codan-LBUC6x00

This page describes the device driver and the device window for the Codan Ku-Band L-Band Blockupconverter Series 6900 transceiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- RX --- -not used-
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	*description*
address	The device driver uses the terminal protocol to talk with the transceiver, hence there is no device addressing. Leave this field empty.
attenuator	This defines the transmit attenuator in dB.
txPowerTresh	This defines the transmit power alarm threshold.

parameter	*description*
burstPowerTresh	This defines the burst power alarm threshold.
txAtPowerUp	If set to 'OFF', the transceiver starts with RF OFF when it is powered up. The 'LAST STATE' setting lets the transceiver return to the state when it was switched off
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. The transceiver either expects *9600,N,8,1* when set to factory settings.

Frequency ranges

The device driver automatically detects the frequency ranges accepted by the transceiver and forwards these limits to the software user interface.

Device reset

The RESET function provided with the 'Maintenance' page issues the following commands to the transceiver:

- reset change bits
- reset latched faults
- reset BUC

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.attenuator	CHOICE	SETUP	EnumRange (0 4 8 12)
config.txPowerTresh	FLOAT	SETUP	DoubleRange (0.0 .. 42.0)

name	*type*	*flags*	*range*
config.burstPowerTresh	FLOAT	SETUP	DoubleRange (0.0 .. 42.0)
config.txAtPowerUp	CHOICE	SETUP	EnumRange (OFF LAST STATE)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.loFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.build	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.minBurst	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
info.maxBurst	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
tx.burstLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
tx.frequency	INTEGER		IntegerRange (0 .. 0)
tx.ifFrequency	INTEGER		IntegerRange (0 .. 0)
tx.on	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.rfmin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.tx.rfmax	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.tx.ifmin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.tx.ifmax	FLOAT	R/O	DoubleRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PA)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (LO unlocked)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.116 Codan-Transceiver

This page describes the device driver and the device window for the Codan 5900 series transceivers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- RX --- This page shows receive part parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	*description*
address	The device driver uses the terminal protocol to talk with the transceiver, hence there is no device addressing. Leave this field empty.
lnbFaults	Enable or disable the recognition of LNB faults by the transceiver. Typically you will have this parameter set to ENABLED if a LNB is connected to the transceiver, DISABLED if there is no LNB.
lowPowerFault	Enable or disable the transmit power monitoring by the transceiver. If ENABLED, the transceiver will generate a 'low output power fault' as soon as the power reading falls below the threshold defined below.
lowPowerThresh	The threshold value for the 'low output power fault'.

parameter	*description*
txDuringWarmup	If set to YES, the transceiver transmits even during the oscillator warmup period. If set to NO, the transceiver inhibits transmission until the oscillator oven reaches it's operation temperature.
txAtPowerUp	If set to 'OFF', the transceiver starts with RF OFF when it is powered up. The 'LAST STATE' setting lets the transceiver return to the state when it was switched off
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. The transceiver either expects *9600,N,8,1* when set to factory settings.

Output power reading

Not all types of SSPA modules which can be used with the transceiver provide an output power reading. If the power reading constantly shows 0.0 dBm, the SSPA probably does not supply the true value.

Frequency ranges

The device driver automatically detects the frequency ranges accepted by the transceiver and forwards these limits to the software user interface.

RF ON/OFF switching

The device driver uses the 'SSPA inhibit' command of the transceiver to implement the RF ON/OFF function. Switching RF ON additionally issues 'SSPA activate' and 'transceiver ON' commands to the device to be sure that these switches also are on.

Device reset

The RESET function provided with the 'Maintenance' page issues the following commands to the transceiver:

- SSPA inhibit ON
- SSPA activate OFF
- transceiver OFF
- Microprocessor RST

The transceiver should be at RF OFF state after the RESET.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.lnbFaults	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.lowPowerFault	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.lowPowerThresh	INTEGER	SETUP	IntegerRange (20 .. 55)
config.txDuringWarmup	CHOICE	SETUP	EnumRange (NO YES)
config.txAtPowerUp	CHOICE	SETUP	EnumRange (OFF LAST STATE)
info.conv.identify	TEXT	R/O	StringRange R/O
info.conv.serial	TEXT	R/O	StringRange R/O
info.conv.interface	TEXT	R/O	StringRange R/O
info.conv.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.sspa.identify	TEXT	R/O	StringRange R/O
info.sspa.serial	TEXT	R/O	StringRange R/O
info.sspa.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.system.on	BOOLEAN	R/O	BooleanRange R/O (true false)
info.sspa.activated	BOOLEAN	R/O	BooleanRange R/O (true false)
info.sspa.inhibited	BOOLEAN	R/O	BooleanRange R/O (true false)
info.if.on	BOOLEAN	R/O	BooleanRange R/O (true false)
info.oven.warmedUp	BOOLEAN	R/O	BooleanRange R/O (true false)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
rx.frequency	FLOAT		DoubleRange (0 .. 0)
rx.gain	FLOAT		DoubleRange (0 .. 25)

name	*type*	*flags*	*range*
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0 .. 0)
tx.gain	FLOAT		DoubleRange (0 .. 25)
tx.on	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.tx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.sspa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.conv.type	TEXT	R/O	StringRange R/O
internal.conv.firmware	TEXT	R/O	StringRange R/O
internal.conv.release	TEXT	R/O	StringRange R/O
internal.conv.version	TEXT	R/O	StringRange R/O
internal.conv.bandwidth	TEXT	R/O	StringRange R/O
internal.sspa.type	TEXT	R/O	StringRange R/O
internal.sspa.firmware	TEXT	R/O	StringRange R/O
internal.sspa.release	TEXT	R/O	StringRange R/O
internal.sspa.version	TEXT	R/O	StringRange R/O
internal.sspa.fan	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.02	ALARM	R/O	AlarmFlagRange R/O (LNB)
faults.03	ALARM	R/O	AlarmFlagRange R/O (SSPA)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Tx-LO lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Rx-LO lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Tx synth 1 lock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Tx synth 2 lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Tx synth 3 lock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Rx synth 1 lock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx synth 2 lock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Bias supply)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Overcurrent)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switched supplies)

9.117 Collector-Two-Way

The logical device Collector-Two-Way defines up to 32 different parameters from other devices (collected into one device) to be monitored and set (two-way). This device is useful to define a set of parameters which are collected together in one device and additionally set them all together. An example is to reduce a collected preset to only necessary parameters and als to set other devices with this reduced preset.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- List of parameters
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	*description*
parameter.xx.Id	Parameter IDs

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.parameter.01.id	TEXT	SAVE SETUP	StringRange
config.parameter.02.id	TEXT	SAVE SETUP	StringRange
config.parameter.03.id	TEXT	SAVE SETUP	StringRange
config.parameter.04.id	TEXT	SAVE SETUP	StringRange
config.parameter.05.id	TEXT	SAVE SETUP	StringRange
config.parameter.06.id	TEXT	SAVE SETUP	StringRange
config.parameter.07.id	TEXT	SAVE SETUP	StringRange
config.parameter.08.id	TEXT	SAVE SETUP	StringRange
config.parameter.09.id	TEXT	SAVE SETUP	StringRange
config.parameter.10.id	TEXT	SAVE SETUP	StringRange
config.parameter.11.id	TEXT	SAVE SETUP	StringRange
config.parameter.12.id	TEXT	SAVE SETUP	StringRange
config.parameter.13.id	TEXT	SAVE SETUP	StringRange
config.parameter.14.id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.parameter.15.id	TEXT	SAVE SETUP	StringRange
config.parameter.16.id	TEXT	SAVE SETUP	StringRange
config.parameter.17.id	TEXT	SAVE SETUP	StringRange
config.parameter.18.id	TEXT	SAVE SETUP	StringRange
config.parameter.19.id	TEXT	SAVE SETUP	StringRange
config.parameter.20.id	TEXT	SAVE SETUP	StringRange
config.parameter.21.id	TEXT	SAVE SETUP	StringRange
config.parameter.22.id	TEXT	SAVE SETUP	StringRange
config.parameter.23.id	TEXT	SAVE SETUP	StringRange
config.parameter.24.id	TEXT	SAVE SETUP	StringRange
config.parameter.25.id	TEXT	SAVE SETUP	StringRange
config.parameter.26.id	TEXT	SAVE SETUP	StringRange
config.parameter.27.id	TEXT	SAVE SETUP	StringRange
config.parameter.28.id	TEXT	SAVE SETUP	StringRange
config.parameter.29.id	TEXT	SAVE SETUP	StringRange
config.parameter.30.id	TEXT	SAVE SETUP	StringRange
config.parameter.31.id	TEXT	SAVE SETUP	StringRange
config.parameter.32.id	TEXT	SAVE SETUP	StringRange
parameter.01	TEXT		StringRange
parameter.02	TEXT		StringRange
parameter.03	TEXT		StringRange
parameter.04	TEXT		StringRange
parameter.05	TEXT		StringRange
parameter.06	TEXT		StringRange
parameter.07	TEXT		StringRange
parameter.08	TEXT		StringRange

name	*type*	*flags*	*range*
parameter.09	TEXT		StringRange
parameter.10	TEXT		StringRange
parameter.11	TEXT		StringRange
parameter.12	TEXT		StringRange
parameter.13	TEXT		StringRange
parameter.14	TEXT		StringRange
parameter.15	TEXT		StringRange
parameter.16	TEXT		StringRange
parameter.17	TEXT		StringRange
parameter.18	TEXT		StringRange
parameter.19	TEXT		StringRange
parameter.20	TEXT		StringRange
parameter.21	TEXT		StringRange
parameter.22	TEXT		StringRange
parameter.23	TEXT		StringRange
parameter.24	TEXT		StringRange
parameter.25	TEXT		StringRange
parameter.26	TEXT		StringRange
parameter.27	TEXT		StringRange
parameter.28	TEXT		StringRange
parameter.29	TEXT		StringRange
parameter.30	TEXT		StringRange
parameter.31	TEXT		StringRange
parameter.32	TEXT		StringRange

9.118 Collector

The logical device Collector monitors up to 32 parameters from other devices. This device is

useful together with the Proxy device, if you want to make read-only parameters available to other MNC systems

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- List of collected parameters
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
input-xx.Id	Parameter IDs

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input-01.Id	TEXT	SAVE SETUP	StringRange
config.input-02.Id	TEXT	SAVE SETUP	StringRange
config.input-03.Id	TEXT	SAVE SETUP	StringRange
config.input-04.Id	TEXT	SAVE SETUP	StringRange
config.input-05.Id	TEXT	SAVE SETUP	StringRange
config.input-06.Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-07.Id	TEXT	SAVE SETUP	StringRange
config.input-08.Id	TEXT	SAVE SETUP	StringRange
config.input-09.Id	TEXT	SAVE SETUP	StringRange
config.input-10.Id	TEXT	SAVE SETUP	StringRange
config.input-11.Id	TEXT	SAVE SETUP	StringRange
config.input-12.Id	TEXT	SAVE SETUP	StringRange
config.input-13.Id	TEXT	SAVE SETUP	StringRange
config.input-14.Id	TEXT	SAVE SETUP	StringRange
config.input-15.Id	TEXT	SAVE SETUP	StringRange
config.input-16.Id	TEXT	SAVE SETUP	StringRange
config.input-17.Id	TEXT	SAVE SETUP	StringRange
config.input-18.Id	TEXT	SAVE SETUP	StringRange
config.input-19.Id	TEXT	SAVE SETUP	StringRange
config.input-20.Id	TEXT	SAVE SETUP	StringRange
config.input-21.Id	TEXT	SAVE SETUP	StringRange
config.input-22.Id	TEXT	SAVE SETUP	StringRange
config.input-23.Id	TEXT	SAVE SETUP	StringRange
config.input-24.Id	TEXT	SAVE SETUP	StringRange
config.input-25.Id	TEXT	SAVE SETUP	StringRange
config.input-26.Id	TEXT	SAVE SETUP	StringRange
config.input-27.Id	TEXT	SAVE SETUP	StringRange
config.input-28.Id	TEXT	SAVE SETUP	StringRange
config.input-29.Id	TEXT	SAVE SETUP	StringRange
config.input-30.Id	TEXT	SAVE SETUP	StringRange
config.input-31.Id	TEXT	SAVE SETUP	StringRange
config.input-32.Id	TEXT	SAVE SETUP	StringRange
config.input-33.Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-34.Id	TEXT	SAVE SETUP	StringRange
config.input-35.Id	TEXT	SAVE SETUP	StringRange
config.input-36.Id	TEXT	SAVE SETUP	StringRange
config.input-37.Id	TEXT	SAVE SETUP	StringRange
config.input-38.Id	TEXT	SAVE SETUP	StringRange
config.input-39.Id	TEXT	SAVE SETUP	StringRange
config.input-40.Id	TEXT	SAVE SETUP	StringRange
config.input-41.Id	TEXT	SAVE SETUP	StringRange
config.input-42.Id	TEXT	SAVE SETUP	StringRange
config.input-43.Id	TEXT	SAVE SETUP	StringRange
config.input-44.Id	TEXT	SAVE SETUP	StringRange
config.input-45.Id	TEXT	SAVE SETUP	StringRange
config.input-46.Id	TEXT	SAVE SETUP	StringRange
config.input-47.Id	TEXT	SAVE SETUP	StringRange
config.input-48.Id	TEXT	SAVE SETUP	StringRange
input.01	TEXT		StringRange
input.02	TEXT		StringRange
input.03	TEXT		StringRange
input.04	TEXT		StringRange
input.05	TEXT		StringRange
input.06	TEXT		StringRange
input.07	TEXT		StringRange
input.08	TEXT		StringRange
input.09	TEXT		StringRange
input.10	TEXT		StringRange
input.11	TEXT		StringRange

name	*type*	*flags*	*range*
input.12	TEXT		StringRange
input.13	TEXT		StringRange
input.14	TEXT		StringRange
input.15	TEXT		StringRange
input.16	TEXT		StringRange
input.17	TEXT		StringRange
input.18	TEXT		StringRange
input.19	TEXT		StringRange
input.20	TEXT		StringRange
input.21	TEXT		StringRange
input.22	TEXT		StringRange
input.23	TEXT		StringRange
input.24	TEXT		StringRange
input.25	TEXT		StringRange
input.26	TEXT		StringRange
input.27	TEXT		StringRange
input.28	TEXT		StringRange
input.29	TEXT		StringRange
input.30	TEXT		StringRange
input.31	TEXT		StringRange
input.32	TEXT		StringRange
input.33	TEXT		StringRange
input.34	TEXT		StringRange
input.35	TEXT		StringRange
input.36	TEXT		StringRange

name	*type*	*flags*	*range*
input.37	TEXT		StringRange
input.38	TEXT		StringRange
input.39	TEXT		StringRange
input.40	TEXT		StringRange
input.41	TEXT		StringRange
input.42	TEXT		StringRange
input.43	TEXT		StringRange
input.44	TEXT		StringRange
input.45	TEXT		StringRange
input.46	TEXT		StringRange
input.47	TEXT		StringRange
input.48	TEXT		StringRange

9.119 ColumbiaWeather-MicroServer

This page describes the device driver and the device window for the Orion Weather station from Columbia Weather Systems Inc. monitored using SNMP via the CWS Micro Server. The driver is intended for monitoring only purpose.

It was tested with model:

- Orion / New Weather MicroServer
- Hardware Model: MARK III
- Firmware Version: MS_4.0.13505

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Measurements
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The Weather Station device needs some configuration parameters as well as the correct settings for the SNMP interface

parameter	***description***
address	The SNMP address specification for the device (details below).
wind.hiThreshold	Defines the high wind limit in m/s which triggers the high wind speed fault.
temp.hiThreshold	Defines the high temperature limit in °Celsius which triggers the high temperature fault.
temp.lothreshold	Defines the low temperature limit in °Celsius which triggers the high wind speed fault.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	**description**
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.lastSample	TEXT	R/O	StringRange R/O
config.wind.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.temp.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.temp.loThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
windchill	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
windchill2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
heatIndex	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
heatIndex2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dewpoint	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dewpoint2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction	INTEGER	R/O	IntegerRange R/O (0 .. 360)
windRose	TEXT	R/O	StringRange R/O
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
pressure3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
speed.3s.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2.3s.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction.3s.average	INTEGER	R/O	IntegerRange R/O (0 .. 360)
speed.2m.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2.2m.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction.2m.average	INTEGER	R/O	IntegerRange R/O (0 .. 360)
speed.10m.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2.10m.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction.10m.average	INTEGER	R/O	IntegerRange R/O (0 .. 360)
speed.10m.gusts	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2.10m.gusts	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction.10m.gusts	INTEGER	R/O	IntegerRange R/O (0 .. 360)
speed.1h.gusts	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed2.1h.gusts	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
direction.1h.gusts	INTEGER	R/O	IntegerRange R/O (0 .. 360)
rain.int	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rain2.int	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
hail.int	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
hail2.int	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rain.day	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rain2.day	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
hail.day	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
hail2.day	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rain.week	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rain2.week	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
rain.month	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rain2.month	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rain.year	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rain2.year	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (High Wind Speed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low Temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High Temperature)

9.120 ComWinTop-THxxS

This page describes the device driver and the device window for the*CWT-THxxS Series RS485 temperature humidity sensor* device. This device reads a temperature and a humidity value.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual temperature and humidity values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device

parameter	*description*
address	Enter the Modbus address of the sensor device here. The factory default is 1

parameter	*description*
Serial	Remember to set the line parameters for the serial interface to*4800,N,8,1* , no handshaking.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.model	CHOICE	SAVE SETUP	EnumRange (CWT-THxxS CWT-XYTH)
internal.temp	FLOAT	R/O	DoubleRange R/O (0.000000000 .. 0.000000000)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

9.121 Comment

The logical device Comment provides a text field which could be used to shows and saves some text. The comment field it lockable by pressing a button which means that you can set it to readonly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the comment and lock button
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	*description*
label	Sets the label of the comment field in the device window. (not required)

Comment and lock status are saved on the M&C system and preserved after a system restart

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.label	TEXT	SAVE SETUP	StringRange
comment	TEXT	SAVE	StringRange
lock	BOOLEAN	SAVE	BooleanRange (true false)

9.122 Compare

The logical device Compare, compares two values from other devices and returns true or false.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	*description*

parameter	*description*
input-A.Id	Parameter ID of first parameter
input-B.Id	Parameter ID of second parameter
compareType	Select between comparison methods

compare methods

- EQ = equal
- NE = not equal
- GT = greater than
- LT = lower than

The result is provided via the state parameter.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.compareType	CHOICE	SAVE SETUP	EnumRange (EQ NE GT LT)
input-A	TEXT		StringRange
input-B	TEXT		StringRange
state	BOOLEAN	R/O	BooleanRange R/O (true false)

9.123 Comstream-ABR200

This page describes the device driver and the device window for the Comstream ABR200 Audio Broadcast Receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the demodulator / decoder settings
- --- Contains the parameters concerning user data / in-band signaling
- --- Shows the actual relay / TTL input status and defines the relay contact sense.
- --- Lets you define the channel and format tables stored in the the receiver. Shows one of the 32 channels and one of the 64 formats at a time. Below find a introduction how to operate this page.
- --- Shows the actual receive state, meter readings like the signal quality and some statistical values recorded by the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the ABR200.

- *parameter --- description*
- address --- If you are using the packet protocol / RS485 you must set this to the packet address set in the ABR200. Valid addresses are 1..31, 31 is the factory default. If you have multiple receivers connected to one RS485 bus, each device must have it's unique address. With the RS232 interface, the packet address doesn't care, you can leave this field empty.
- login --- If controlled via the RS232, the ABR200 requires a login password to be entered before the receiver can be controlled. Enter this password here, the software sends it to the receiver each time it looses communication to the device. The factory preset for the login password at the ABR200 is "HOMEYD". When communicating in packet mode through the RS485, the ABR200 doesn't request (and also doesn't accept) a password. Set this field to "RS485" in this case. This prevents the software from trying to log in at the ABR200.
- Serial --- Remember to set the line parameters for the serial interface connected to the ABR200. While the ABR200's factory preset setting is *2400,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* . See below how to change this setting at the ABR200.

Defining channels / formats

The channel / format screen of the device window shows one channel and one format definition definition at a time. To enter a new channel definition or to edit an existing one, do the following steps:

1. Select the channel to add / edit at the "channel" field.
2. Ensure that the "enabled" field is set "ON".
3. Enter the parameters for this channel If you enter channel parameters while the "enabled" parameter is off, the ABR will not recognize the changes you make, so be sure to enable the channel before you set the frequency, data rate etc.

To delete a channel definition, simply set the "enable" parameter to "OFF".

Format definitions are added/edited/deleted accordingly by means of the fields shown in the lower part in this screen.

Communication protocol / Interface type

The recommended communication protocol for the ABR200 is the *Comstream-Packet-Protocol* . You must connect at the receiver's RS485 M&C port to use this protocol type. If you want to control the ABR200 by RS232 for some reason, choose the *Comstream-ASCII-Protocol* when you configure the device setup.

Installation

The ABR200 does not have any front panel control. In order to prepare the ABR200 for use with the satnms software, connect an ASCII terminal (or a PC running a terminal program) to the ABR200. If you have a 9 pin RS232 connector at your terminal, you need a straight through 9-pin cable (male/female) for this. The ABR200's factory line settings are 2400,O,7,1, first try this setting at your terminal. The login password is "HOMEYD" if the receiver comes out of the box. Now command the following settings:

command	***description***
PA 31	Sets the receivers packet address. You may choose any value between 1 and 31 as long as you set the same value at the *address* configuration parameter in the software. If you are operating multiple receivers on one common interface, ensure that each receiver is set to a unique address.
P2 9600,O,7,1	Sets the receivers M&C port to the recommended line settings. Your terminal may show strange characters after you entered this, because it does not understand the data sent by the ABR200 unless you change the line settings at the terminal, too.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.id	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (AGC range)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Bit time lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Carrier track lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (FEC decoder sync)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Acquisition range)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Carrier track range)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Bit time range)
faults.08	ALARM	R/O	AlarmFlagRange R/O (NV RAM)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Watchdog timer)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Audio PLL lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Audio decoder sync)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DSP watchdog)
faults.13	ALARM	R/O	AlarmFlagRange R/O (DSP bit failure)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Extern input 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Extern input 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Extern input 6)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Outdoor unit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Eb/No threshold)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EPROM checksum)
faults.20	ALARM	R/O	AlarmFlagRange R/O (S/W download)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Channel change)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Network ID timeout)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Acquisition network ID)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ch.frequency	FLOAT	R/O	DoubleRange R/O (10950.000 .. 12750.000)
internal.ch.rate	CHOICE	R/O	EnumRange R/O (64000 96000 128000 192000 256000 384000 512000)
internal.ch.modType	CHOICE	R/O	EnumRange R/O (BPSK QPSK)
internal.ch.fec	CHOICE	R/O	EnumRange R/O (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 2/3-VIT 3/4-VIT 5/6-VIT 7/8-VIT)
internal.ch.block	CHOICE	R/O	EnumRange R/O (188/204 187/204)
internal.fm.networkID	INTEGER	R/O	IntegerRange R/O (0 .. 255)
internal.fm.channelID	INTEGER	R/O	IntegerRange R/O (0 .. 31)
internal.fm.ibsSelect	CHOICE	R/O	EnumRange R/O (OFF AUDIO DATA A+D RELAY A+R A+D ALL)
internal.fm.lrToggle	CHOICE	R/O	EnumRange R/O (NORMAL REVERSE LEFT RIGHT)
internal.fm.mute	CHOICE	R/O	EnumRange R/O (UNMUTE MUTE MUTE-L MUTE-R)
config.login	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultRelay	CHOICE	SETUP	EnumRange (NRM INV)
reset	TEXT	NOPRESET	StringRange
acquisition	CHOICE		EnumRange (STOP FADE POWER-ON CH-SRCH)
testTone	CHOICE		EnumRange (OFF ON)
lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)
decoderSource	CHOICE		EnumRange (DEMOD AUTO EXTERNAL)
spectralInversion	CHOICE		EnumRange (NRM INV)
mute	CHOICE		EnumRange (UNMUTE MUTE MUTE-L MUTE-R)
volume	INTEGER		IntegerRange (0 .. 18)
networkData	CHOICE		EnumRange (OFF USER PRINTER PASSTHRU)
compositeData	CHOICE		EnumRange (OFF ON)
meter.agc	INTEGER	R/O	IntegerRange R/O (0 .. 255)
meter.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 12750.000)
meter.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.dataRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.ber	TEXT	R/O	StringRange R/O
meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 20.0)
meter.minEbno	FLOAT	R/O	DoubleRange R/O (0.0 .. 22.0)
meter.maxEbno	FLOAT	R/O	DoubleRange R/O (0.0 .. 20.0)
meter.fades	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.syncLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.clear	TEXT		StringRange
networkStatus	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
threshold.ebno	FLOAT		DoubleRange (0.1 .. 20.0)
threshold.muteOn	FLOAT		DoubleRange (0.1 .. 20.0)
threshold.muteOff	FLOAT		DoubleRange (0.1 .. 20.0)
binning.range1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
binning.range2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
binning.range3	INTEGER		IntegerRange (0 .. 4000)
channel.edit	INTEGER		IntegerRange (0 .. 31)
channel.enable	CHOICE		EnumRange (OFF ON)
channel.frequency	FLOAT		DoubleRange (10950.000 .. 12750.000)
channel.rate	CHOICE		EnumRange (64000 96000 128000 192000 256000 384000 512000)
channel.modType	CHOICE		EnumRange (BPSK QPSK)
channel.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 2/3-VIT 3/4-VIT 5/6-VIT 7/8-VIT)
channel.block	CHOICE		EnumRange (188/204 187/204)
contactSense.1	CHOICE		EnumRange (NRM INV)
contactSense.2	CHOICE		EnumRange (NRM INV)
contactSense.3	CHOICE		EnumRange (NRM INV)
contactSense.4	CHOICE		EnumRange (NRM INV)
contactSense.5	CHOICE		EnumRange (NRM INV)
contactSense.6	CHOICE		EnumRange (NRM INV)
contactSense.7	CHOICE		EnumRange (NRM INV)
contactSense.8	CHOICE		EnumRange (NRM INV)
contact.1	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.2	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.3	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.4	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
contact.5	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.6	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.7	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.8	CHOICE	R/O	EnumRange R/O (OFF ON)
ttl.1	TEXT	R/O	StringRange R/O
ttl.2	TEXT	R/O	StringRange R/O
ttl.3	TEXT	R/O	StringRange R/O
ttl.4	TEXT	R/O	StringRange R/O
ttl.5	TEXT	R/O	StringRange R/O
ttl.6	TEXT	R/O	StringRange R/O
ttl.7	TEXT	R/O	StringRange R/O
format.edit	INTEGER		IntegerRange (0 .. 63)
format.enable	CHOICE		EnumRange (OFF ON)
format.networkID	INTEGER		IntegerRange (0 .. 255)
format.channelID	INTEGER		IntegerRange (0 .. 31)
format.ibsSelect	CHOICE		EnumRange (OFF AUDIO DATA A+D RELAY A+R A+D ALL)
format.lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)
format.mute	CHOICE		EnumRange (UNMUTE MUTE MUTE-L MUTE-R)
format.select	INTEGER		IntegerRange (0 .. 63)
data.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)
data.parity	CHOICE		EnumRange (O E N)
data.databits	CHOICE		EnumRange (7 8)
data.stopbits	CHOICE		EnumRange (1 2)
printer.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)

name	*type*	*flags*	*range*
printer.parity	CHOICE		EnumRange (O E N)
printer.databits	CHOICE		EnumRange (7 8)
printer.stopbits	CHOICE		EnumRange (1 2)
printer.interface	CHOICE		EnumRange (RS232 RS422)
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

9.124 Comstream-ABR202

This page describes the device driver and the device window for the Comstream ABR202 Audio Broadcast Receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the demodulator / decoder settings
- --- Contains the parameters concerning user data / in-band signaling
- --- Shows the actual relay / TTL input status and defines the relay contact sense.
- --- Lets you define the channel and format tables stored in the the receiver. Shows one of the 32 channels and one of the 64 formats at a time. Below find a introduction how to operate this page.
- --- Shows the actual receive state, meter readings like the signal quality and some statistical values recorded by the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the ABR202.

- *parameter --- description*
- address --- If you are using the packet protocol / RS485 you must set this to the packet address set in the ABR202. Valid addresses are 1..31, 31 is the factory default. If you

have multiple receivers connected to one RS485 bus, each device must have it's unique address. With the RS232 interface, the packet address doesn't care, you can leave this field empty.

- login --- If controlled via the RS232, the ABR202 requires a login password to be entered before the receiver can be controlled. Enter this password here, the software sends it to the receiver each time it looses communication to the device. The factory preset for the login password at the ABR202 is "HOMEYD".When communicating in packet mode through the RS485, the ABR202 doesn't request (and also doesn't accept) a password. Set this field to "RS485" in this case. This prevents the software from trying to log in at the ABR202.
- Serial --- Remember to set the line parameters for the serial interface connected to the ABR202. While the ABR202's factory preset setting is *2400,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* . See below how to change this setting at the ABR202.

Defining channels / formats

The cannel / format screen of the device window shows one channel and one format definition definition at a time. To enter a new channel definition or to edit an existing one, do the following steps:

1. Select the channel to add / edit at the "channel" field.
2. Ensure that the "enabled" field is set "ON".
3. Enter the parameters for this channel If you enter channel parameters while the "enabled" parameter is off, the ABR will not recognize the changes you make, so be sure to enable the channel before you set the frequency, data rate etc.

To delete a channel definition, simply set the "enable" parameter to "OFF".

Format definitions are added/edited/deleted accordingly by means of the fields shown in the lower part in this screen.

Communication protocol / Interface type

The recommended communication protocol for the ABR202 is the*Comstream-Packet-Protocol* . You must connect at the receiver's RS485 M&C port to use this protocol type. If you want to control the ABR202 by RS232 for some reason, choose the *Comstream-ASCII-Protocol* when you configure the device setup.

Installation

The ABR202 does not have any front panel control. In order to prepare the ABR202 for use with the satnms software, connect an ASCII terminal (or a PC running a terminal program) to the ABR202. If you have a 9 pin RS232 connector at your terminal, you need a straight through 9-pin cable (male/female) for this. The ABR202's factory line settings are 2400,O,7,1, first try this setting at your terminal. The login password is "HOMEYD" if the receiver comes out of the box. Now command the following settings:

command	***description***

command	*description*
PA 31	Sets the receivers packet address. You may choose any value between 1 and 31 as long as you set the same value at the *address* configuration parameter in the software. If you are operating multiple receivers on one common interface, ensure that each receiver is set to a unique address.
P2 9600,O,7,1	Sets the receivers M&C port to the recommended line settings. Your terminal may show strange characters after you entered this, because it does not understand the data sent by the ABR202 unless you change the line settings at the terminal, too.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.id	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (AGC range)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Bit time lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Carrier track lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (FEC decoder sync)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Acquisition range)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Carrier track range)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Bit time range)
faults.08	ALARM	R/O	AlarmFlagRange R/O (NV RAM)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Watchdog timer)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Audio PLL lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Audio decoder sync)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DSP watchdog)
faults.13	ALARM	R/O	AlarmFlagRange R/O (DSP bit failure)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Extern input 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Extern input 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Extern input 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Outdoor unit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Eb/No threshold)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EPROM checksum)
faults.20	ALARM	R/O	AlarmFlagRange R/O (S/W download)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Channel change)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Network ID timeout)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Acquisition network ID)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ch.frequency	FLOAT	R/O	DoubleRange R/O (10950.000 .. 12750.000)
internal.ch.rate	CHOICE	R/O	EnumRange R/O (64000 96000 128000 192000 256000 384000 512000)
internal.ch.modType	CHOICE	R/O	EnumRange R/O (BPSK QPSK)
internal.ch.fec	CHOICE	R/O	EnumRange R/O (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 2/3-VIT 3/4-VIT 5/6-VIT 7/8-VIT)
internal.ch.block	CHOICE	R/O	EnumRange R/O (188/204 187/204)
internal.fm.networkID	INTEGER	R/O	IntegerRange R/O (0 .. 255)

name	*type*	*flags*	*range*
internal.fm.channelID	INTEGER	R/O	IntegerRange R/O (0 .. 31)
internal.fm.ibsSelect	CHOICE	R/O	EnumRange R/O (OFF AUDIO DATA A+D RELAY A+R A+D ALL)
internal.fm.lrToggle	CHOICE	R/O	EnumRange R/O (NORMAL REVERSE LEFT RIGHT)
internal.fm.mute	CHOICE	R/O	EnumRange R/O (UNMUTE MUTE MUTE-L MUTE-R)
config.login	TEXT	SAVE SETUP	StringRange
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultRelay	CHOICE	SETUP	EnumRange (NRM INV)
reset	TEXT	NOPRESET	StringRange
acquisition	CHOICE		EnumRange (STOP FADE POWER-ON CH-SRCH)
testTone	CHOICE		EnumRange (OFF ON)
lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)
decoderSource	CHOICE		EnumRange (DEMOD AUTO EXTERNAL)
spectralInversion	CHOICE		EnumRange (NRM INV)
mute	CHOICE		EnumRange (UNMUTE MUTE MUTE-L MUTE-R)
volume	INTEGER		IntegerRange (0 .. 18)
networkData	CHOICE		EnumRange (OFF USER PRINTER PASSTHRU)
compositeData	CHOICE		EnumRange (OFF ON)
meter.agc	INTEGER	R/O	IntegerRange R/O (0 .. 255)
meter.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 12750.000)
meter.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.dataRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meter.ber	TEXT	R/O	StringRange R/O
meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 20.0)
meter.minEbno	FLOAT	R/O	DoubleRange R/O (0.0 .. 22.0)
meter.maxEbno	FLOAT	R/O	DoubleRange R/O (0.0 .. 20.0)
meter.fades	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.syncLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.clear	TEXT		StringRange
networkStatus	TEXT	R/O	StringRange R/O
threshold.ebno	FLOAT		DoubleRange (0.1 .. 20.0)
threshold.muteOn	FLOAT		DoubleRange (0.1 .. 20.0)
threshold.muteOff	FLOAT		DoubleRange (0.1 .. 20.0)
binning.range1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
binning.range2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
binning.range3	INTEGER		IntegerRange (0 .. 4000)
channel.edit	INTEGER		IntegerRange (0 .. 31)
channel.enable	CHOICE		EnumRange (OFF ON)
channel.frequency	FLOAT		DoubleRange (10950.000 .. 12750.000)
channel.rate	CHOICE		EnumRange (64000 96000 128000 192000 256000 384000 512000)
channel.modType	CHOICE		EnumRange (BPSK QPSK)
channel.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 2/3-VIT 3/4-VIT 5/6-VIT 7/8-VIT)
channel.block	CHOICE		EnumRange (188/204 187/204)
contactSense.1	CHOICE		EnumRange (NRM INV)
contactSense.2	CHOICE		EnumRange (NRM INV)
contactSense.3	CHOICE		EnumRange (NRM INV)
contactSense.4	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
contactSense.5	CHOICE		EnumRange (NRM INV)
contactSense.6	CHOICE		EnumRange (NRM INV)
contactSense.7	CHOICE		EnumRange (NRM INV)
contactSense.8	CHOICE		EnumRange (NRM INV)
contact.1	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.2	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.3	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.4	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.5	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.6	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.7	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.8	CHOICE	R/O	EnumRange R/O (OFF ON)
ttl.1	TEXT	R/O	StringRange R/O
ttl.2	TEXT	R/O	StringRange R/O
ttl.3	TEXT	R/O	StringRange R/O
ttl.4	TEXT	R/O	StringRange R/O
ttl.5	TEXT	R/O	StringRange R/O
ttl.6	TEXT	R/O	StringRange R/O
ttl.7	TEXT	R/O	StringRange R/O
format.edit	INTEGER		IntegerRange (0 .. 63)
format.enable	CHOICE		EnumRange (OFF ON)
format.networkID	INTEGER		IntegerRange (0 .. 255)
format.channelID	INTEGER		IntegerRange (0 .. 31)
format.ibsSelect	CHOICE		EnumRange (OFF AUDIO DATA A+D RELAY A+R A+D ALL)
format.lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)

name	*type*	*flags*	*range*
format.mute	CHOICE		EnumRange (UNMUTE MUTE MUTE-L MUTE-R)
format.select	INTEGER		IntegerRange (0 .. 63)
data.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)
data.parity	CHOICE		EnumRange (O E N)
data.databits	CHOICE		EnumRange (7 8)
data.stopbits	CHOICE		EnumRange (1 2)
printer.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)
printer.parity	CHOICE		EnumRange (O E N)
printer.databits	CHOICE		EnumRange (7 8)
printer.stopbits	CHOICE		EnumRange (1 2)
printer.interface	CHOICE		EnumRange (RS232 RS422)
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

9.125 Comstream-CM601A-SCPC

This page describes the device driver and the device window for the Comstream CM601A PSK Satellite Modem. In future, there will be several variants of this device driver to be used with different modem configurations. Actually the *Comstream-CM601A-SCPC* driver variant is available only.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The TX interface settings.
- --- The RX interface settings.

- --- The receive quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the CM601A.

- *parameter --- description*
- address --- The modem's RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address. If you want to access a CM601A through a CX801/CX101 protection switch, you have to prepend the address of the CX801/CX101 separated by a comma: "1,15" addresses the modem "15" through the protection switch "1".
- type --- Select if the modem is equipped with a standard 512 kBit modem card or with a high speed modem card (5 MBit).
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask".
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the CM601. While the CM601's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM601 is the*Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Installation

Before the sat-nms software will recognize the modem, you have to set the communication parameters at the modem's front panel. The following values must be set (all in the "M&C" menu):

- 9600 baud
- 7 bit / odd
- packet protocol
- packet address 1..31
- interface RS232/RS485

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.type	CHOICE	SAVE SETUP	EnumRange (512 kBit 5 MBit)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.revision.sl4	TEXT	R/O	StringRange R/O
info.revision.sl5	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
modemType	CHOICE		EnumRange (SCPC)
refClkSrc	CHOICE		EnumRange (INT)
loopBack	CHOICE		EnumRange (OFF IF NEAR FAR)
tx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.hardware	CHOICE		EnumRange (V.35 RS-530 RS-232 NONE)
tx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
rx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 512000)

name	*type*	*flags*	*range*
rx.dem.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.clockSource	CHOICE		EnumRange (EXTERNAL EXT-TTL SATELLITE INTERNAL)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.buf.size	INTEGER		IntegerRange (64 .. 262144)
rx.buf.mode	CHOICE		EnumRange (OFF ON)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (-100 .. -60)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 100)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.hardware	CHOICE	R/O	EnumRange R/O (V.35 RS-530 RS-232 NONE)
rx.ifc.handshake	CHOICE	R/O	EnumRange R/O (IGNORE RESPOND)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.02	ALARM	R/O	AlarmFlagRange R/O (TxIF synth)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RxIF synth)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Tx bit timing)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Rx AGC range)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Bit time sync)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Carr track sync)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Decoder sync)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Acquisition)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Rx carr range)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RxBT range)
faults.12	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx buffer overflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Rx buffer underflow)
faults.15	ALARM	R/O	AlarmFlagRange R/O (M&C)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 3)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 4)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Doppler output clock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 5)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Narr option Tx summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Narr option Rx summary)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Narr option Tx hardware)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Narr option Rx hardware)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Tx FIFO full)

name	*type*	*flags*	*range*
faults.27	ALARM	R/O	AlarmFlagRange R/O (Low Eb/No)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Modem card Tx hardware)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Modem card Rx hardware)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dm	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.126 Comstream-CM701-MC

This page describes the device driver and the device window for the Comstream CM701 Satellite Modem for configurations using a single 5MBit / 512kBit Modem card and - optionally - a framer/buffer card and a Reed-Solomon card. This driver does not work with the old type of the lo-speed modem card.

Although this driver is capable to handle a CM701 equipped with a single modem card in a SCPC configuration, it is optimized for applications which make use of the framer option board or the Reed-Solomon codec. The driver does not allow to have an additional modem card, modulator or demodulator installed beside the primary card as it does not use slot addressing with the commands sent to the modem.

The driver automatically recognizes the presence of a framer/buffer or a Reed-Solomon card and enables the parameter settings belonging to these option cards at the user interface. For your convenience, the modem driver uses the "Primary Data Rate" macros of the CM701. These macros automatically compute and set the input/output data rates for the framer and the Reed-Solomon option modules when you enter a data rate value.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a

modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	***group***	***description***
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The TX interface settings.
- --- The Reed-Solomon settings.
- --- The framer option board settings.
- --- The RX interface settings.
- --- The receive quality.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the CM701.

- *parameter --- description*
- address --- The modem's RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address. If you want to access a CM701 through a CX801/CX101 protection switch, you have to prepend the address of the CX801/CX101 separated by a comma: "1,15" addresses the modem "15" through the protection switch "1".
- type --- Adjust this parameter to the data rate range of the modem card (512kBit / 5MBit). *faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask".
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the CM601. While the CM601's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM701 is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *The data rate field at the user interface contains no value. What happened?* --- The primary data rate macros used to program the modem do not work in both directions. If a data rate is set, the CM701 remembers this value and reports it to the software as the

actual setting. This setting is cleared if any intermediate data rate value is changed (e.g. the symbol rate). After power on, the CM701 does not report a valid data rate.

- *One of the IF frequency fields at the user interface contains no value. Is this a similar issue?* --- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.
- *What do the "set to xxx but reported yyy" messages in the event log mean?*-- The range of some parameters of the CM701 depend on other parameter settings. For instance, the Reed-Solomon scrambling type only can be selected, if the Reed-Solomon codec is present and enabled. If not, the CM701 reverts to the old setting an one of these messages appears in the event log. Another parameter commonly causing such messages is the doppler buffer size. The value always must be a power of two, the CM701 automatically uses the next possible value above if you enter some other numbers.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.type	CHOICE	SAVE SETUP	EnumRange (512 kBit 5 MBit)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)

name	*type*	*flags*	*range*
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.slot.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.slot.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.slot.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.slot.modem	TEXT	R/O	StringRange R/O
info.slot.rs	TEXT	R/O	StringRange R/O
info.slot.framer	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
modemType	CHOICE		EnumRange (SCPC BUFFER IDR IBS E1 D&I T1 D&I)
refClkSrc	CHOICE		EnumRange (INT)
loopBack	CHOICE		EnumRange (OFF IF NEAR FAR)
duplexMode	CHOICE		EnumRange (DUPLEX TX-ONLY RX-ONLY)
tx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.symbolRate	INTEGER	R/O	IntegerRange R/O (4800 .. 512000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR IBS R/S IDR-AUTO I-R/S-AUTO I-R/S-MAN)
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.hardware	CHOICE		EnumRange (V.35 RS-530 RS-232 NONE)
tx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
rx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.symbolRate	INTEGER	R/O	IntegerRange R/O (4800 .. 512000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.clockSource	CHOICE		EnumRange (TT-CLK EXT-CONN RT-CLK ST-CLK)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR IBS R/S IDR-AUTO I-R/S-AUTO I-R/S-MAN)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.buf.size	INTEGER		IntegerRange (64 .. 262144)

name	*type*	*flags*	*range*
rx.buf.mode	CHOICE		EnumRange (OFF ON)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.hardware	CHOICE	R/O	EnumRange R/O (V.35 RS-530 RS-232 NONE)
rx.ifc.handshake	CHOICE	R/O	EnumRange R/O (IGNORE RESPOND)
tx.rs.bypass	CHOICE		EnumRange (OFF TX RX RX+TX)
tx.rs.interleave	INTEGER		IntegerRange (1 .. 16)
tx.rs.wordSize	INTEGER		IntegerRange (1 .. 239)
rx.rs.bypass	CHOICE	R/O	EnumRange R/O (OFF TX RX RX+TX)
rx.rs.interleave	INTEGER		IntegerRange (1 .. 16)
rx.rs.wordSize	INTEGER		IntegerRange (1 .. 239)
rx.rs.ber	TEXT	R/O	StringRange R/O
tx.frm.t1FrameFormat	CHOICE		EnumRange (T1ESF T1D4)
tx.frm.dropSlots1	TEXT		StringRange
tx.frm.dropSlots2	TEXT		StringRange
tx.frm.dropSlots3	TEXT		StringRange
tx.frm.dropSlots4	TEXT		StringRange
tx.frm.idrOvhEnable	CHOICE		EnumRange (OFF ON)
tx.frm.idrOvhMode	CHOICE		EnumRange (AUDIO DATA)
tx.frm.audioGain	CHOICE		EnumRange (-25 dB -21 dB -17 dB -12 dB -9 dB -3 dB +1 dB +5 dB)
tx.frm.e1IbsSignaling	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
rx.frm.e1EnableCrc	CHOICE		EnumRange (OFF ON)
rx.frm.insertSlots1	TEXT		StringRange
rx.frm.insertSlots2	TEXT		StringRange
rx.frm.insertSlots3	TEXT		StringRange
rx.frm.insertSlots4	TEXT		StringRange
rx.frm.audioGain	CHOICE		EnumRange (-25 dB -21 dB -17 dB -12 dB -9 dB -3 dB +1 dB +5 dB)
rx.frm.e1IbsSignaling	CHOICE		EnumRange (OFF ON)
rx.frm.frameAlignment	CHOICE		EnumRange (3 Frames 4 Frames)
rx.frm.relayEnable	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.02	ALARM	R/O	AlarmFlagRange R/O (TxIF synth)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RxIF synth)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Tx bit timing)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Rx AGC range)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Bit time sync)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Carr track sync)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Decoder sync)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Acquisition)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Rx carr range)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RxBT range)
faults.12	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx buffer overflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Rx buffer underflow)
faults.15	ALARM	R/O	AlarmFlagRange R/O (M&C)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 3)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 4)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Doppler output clock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 5)
faults.20	ALARM	R/O	AlarmFlagRange R/O (V. option Tx summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (V. option Rx summary)
faults.23	ALARM	R/O	AlarmFlagRange R/O (V. option Tx hardware)
faults.24	ALARM	R/O	AlarmFlagRange R/O (V. option Rx hardware)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Tx FIFO full)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Low Eb/No)
faults.30	ALARM	R/O	AlarmFlagRange R/O (H. option Tx hardware)
faults.31	ALARM	R/O	AlarmFlagRange R/O (H. option Rx hardware)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #1)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #2)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #3)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #4)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #1)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #2)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #3)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #4)

name	*type*	*flags*	*range*
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stx	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.127 Comstream-CM701

This page describes the device driver and the device window for the Comstream CM701 Satellite Modem for configurations using a modulator /demodulator card and - optionally - a framer/buffer card and a Reed-Solomon card.

The driver expects the cards at fixed slot positions as shown below:

Slot	*Card*
1	empty
2	Data I/O Card
3	Buffer Card (optional)
4	M&C Card
5	Buffer/Framer or Reed-Solomon Card (optional)
6	Demodulator Card (70 MHz/140 MHz/L-Band)
7	Modulator Card (70 MHz/140 MHz)

The driver automatically recognizes which cards are available and enables the appropriate parameters at the user interface. It rawly supports tx-only or rx-only configurations, too. The modem may however reply with an "ER ?" to some command settings in this case.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a

modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	***group***	***description***
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The TX interface settings.
- --- The Reed-Solomon settings.
- --- The framer option board settings.
- --- The RX interface settings.
- --- The receive quality.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the CM701.

- *parameter --- description*
- address --- The modem's RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address. If you want to access a CM701 through a CX801/CX101 protection switch, you have to prepend the address of the CX801/CX101 separated by a comma: "1,15" addresses the modem "15" through the protection switch "1".
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask".
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the CM601. While the CM601's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM701 is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *The data rate field at the user interface contains no value. What happened?*--- The primary data rate macros used to program the modem do not work in both directions. If a data rate is set, the CM701 remembers this value and reports it to the software as the actual setting. This setting is cleared if any intermediate data rate value is changed (e.g.

the symbol rate). After power on, the CM701 does not report a valid data rate. The Comstream-CM701 device driver uses the primary data rate macros as soon as it detects the presence of a Buffer/Framer Card or a Reed-Solomon Card. It uses the macros even if the modem is configured to work in SCPC mode without any overhead.

- *One of the IF frequency fields at the user interface contains no value. Is this a similar issue?* --- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.
- *What do the "set to xxx but reported yyy" messages in the event log mean?* --- The range of some parameters of the CM701 depend on other parameter settings. For instance, the Reed-Solomon scrambling type only can be selected, if the Reed-Solomon codec is present and enabled. If not, the CM701 reverts to the old setting an one of these messages appears in the event log. Another parameter commonly causing such messages is the doppler buffer size. The value always must be a power of two, the CM701 automatically uses the next possible value above if you enter some other numbers.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)

name	*type*	*flags*	*range*
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.rsFound	TEXT	R/O	StringRange R/O
info.framerFound	TEXT	R/O	StringRange R/O
info.bufferFound	TEXT	R/O	StringRange R/O
info.modFound	TEXT	R/O	StringRange R/O
info.demFound	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
modemType	CHOICE		EnumRange (SCPC BUFFER IDR IBS E1 D&I T1 D&I)
refClkSrc	CHOICE		EnumRange (INT)
loopBack	CHOICE		EnumRange (OFF IF NEAR FAR)
tx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 2340000)
tx.mod.symbolRate	INTEGER	R/O	IntegerRange R/O (4800 .. 2340000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NO 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR IBS R/S IDR-AUTO I-R/S-AUTO I-R/S-MAN)

name	*type*	*flags*	*range*
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.handshake	CHOICE	R/O	EnumRange R/O (IGNORE RESPOND)
tx.ifc.hardware	TEXT	R/O	StringRange R/O
rx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 2340000)
rx.dem.symbolRate	INTEGER	R/O	IntegerRange R/O (4800 .. 2340000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NO 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.clockSource	CHOICE		EnumRange DIS (TT-CLK RT-CLK)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR IBS R/S IDR-AUTO I-R/S-AUTO I-R/S-MAN)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.buf.size	INTEGER		IntegerRange DIS (32 .. 131072)
rx.buf.mode	CHOICE		EnumRange DIS (OFF ON)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (0 .. 255)
rx.meter.ber	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
rx.ifc.hardware	TEXT	R/O	StringRange R/O
tx.rs.bypass	CHOICE		EnumRange (OFF TX RX RX+TX)
tx.rs.interleave	INTEGER		IntegerRange (1 .. 16)
tx.rs.wordSize	INTEGER		IntegerRange (1 .. 239)
rx.rs.bypass	CHOICE	R/O	EnumRange R/O (OFF TX RX RX+TX)
rx.rs.interleave	INTEGER		IntegerRange (1 .. 16)
rx.rs.wordSize	INTEGER		IntegerRange (1 .. 239)
rx.rs.ber	TEXT	R/O	StringRange R/O
tx.frm.t1FrameFormat	CHOICE		EnumRange (T1ESF T1D4)
tx.frm.dropSlots1	TEXT		StringRange
tx.frm.dropSlots2	TEXT		StringRange
tx.frm.dropSlots3	TEXT		StringRange
tx.frm.dropSlots4	TEXT		StringRange
tx.frm.idrOvhEnable	CHOICE		EnumRange (OFF ON)
tx.frm.idrOvhMode	CHOICE		EnumRange (AUDIO DATA)
tx.frm.audioGain	CHOICE		EnumRange (-25 dB -21 dB -17 dB -12 dB -9 dB -3 dB +1 dB +5 dB)
tx.frm.e1IbsSignaling	CHOICE		EnumRange (OFF ON)
rx.frm.e1EnableCrc	CHOICE		EnumRange (OFF ON)
rx.frm.insertSlots1	TEXT		StringRange
rx.frm.insertSlots2	TEXT		StringRange
rx.frm.insertSlots3	TEXT		StringRange

name	*type*	*flags*	*range*
rx.frm.insertSlots4	TEXT		StringRange
rx.frm.audioGain	CHOICE		EnumRange (-25 dB -21 dB -17 dB -12 dB -9 dB -3 dB +1 dB +5 dB)
rx.frm.e1IbsSignaling	CHOICE		EnumRange (OFF ON)
rx.frm.frameAlignment	CHOICE		EnumRange (3 Frames 4 Frames)
rx.frm.relayEnable	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.02	ALARM	R/O	AlarmFlagRange R/O (TxIF synth)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RxIF synth)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Tx bit timing)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Rx AGC range)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Bit time sync)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Carr track sync)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Decoder sync)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Acquisition)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Rx carr range)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RxBT range)
faults.12	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx buffer overflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Rx buffer underflow)
faults.15	ALARM	R/O	AlarmFlagRange R/O (M&C)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 3)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 4)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Doppler output clock)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Relay monitor 5)
faults.20	ALARM	R/O	AlarmFlagRange R/O (V. option Tx summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (V. option Rx summary)
faults.23	ALARM	R/O	AlarmFlagRange R/O (V. option Tx hardware)
faults.24	ALARM	R/O	AlarmFlagRange R/O (V. option Rx hardware)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Tx FIFO full)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Low Eb/No)
faults.30	ALARM	R/O	AlarmFlagRange R/O (H. option Tx hardware)
faults.31	ALARM	R/O	AlarmFlagRange R/O (H. option Rx hardware)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #1)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #2)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #3)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #4)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #1)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #2)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #3)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #4)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.x	TEXT	R/O	StringRange R/O
internal.p	TEXT	R/O	StringRange R/O

9.128 Comstream-CX101

There is no special online help for this type of device. Please refer to the general description of device windows instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Watchdog)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Program checksum)
internal.ss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pe	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.fs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
automatic	CHOICE		EnumRange (MANUAL AUTOMATIC)
demTimeout	INTEGER		IntegerRange (2 .. 255)
modTimeout	FLOAT		DoubleRange (0.0 .. 9.9)
modemTiming	CHOICE		EnumRange (EXTERNAL INTERNAL)
monitoring	CHOICE		EnumRange (MOD/DEM MOD MOD-HW DEM DEM-HW MOD-HW/DEM MOD/DEM-HW MOD-HW/DEM-HW)
shadow	TEXT		StringRange
spare	CHOICE		EnumRange (1 2 3 4 5 6 7 8 NONE)
protect.1	CHOICE		EnumRange (OFF ON)
protect.2	CHOICE		EnumRange (OFF ON)
protect.3	CHOICE		EnumRange (OFF ON)
protect.4	CHOICE		EnumRange (OFF ON)
protect.5	CHOICE		EnumRange (OFF ON)
protect.6	CHOICE		EnumRange (OFF ON)
protect.7	CHOICE		EnumRange (OFF ON)
protect.8	CHOICE		EnumRange (OFF ON)
enable.1	CHOICE		EnumRange (OFF ON)
enable.2	CHOICE		EnumRange (OFF ON)
enable.3	CHOICE		EnumRange (OFF ON)
enable.4	CHOICE		EnumRange (OFF ON)
enable.5	CHOICE		EnumRange (OFF ON)
enable.6	CHOICE		EnumRange (OFF ON)
enable.7	CHOICE		EnumRange (OFF ON)
enable.8	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
enable.S	CHOICE		EnumRange (OFF ON)
modemFault.1	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.2	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.3	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.4	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.5	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.6	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.7	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.8	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.S	CHOICE	R/O	EnumRange R/O (OK FLT)

9.129 Comstream-CX801

This page describes the device driver and the device window for the Comstream CX801 Modem Protection Switch. The onhand device driver requires the CX801 to be controlled via RS485, the RS232 version of the CX801 is not supported.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational equipment settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the CX801.

parameter	*description*
address	The RS485 packet address (1..31). The value must match the address set at the CX801 rear panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the CX801. The recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Installation

In order to make the CX801 talk to the M&C computer / VLC, you must do the following settings at the CX801:

- Set the packet address at the dip switches at the CX801's back to the device address you want to use.
- Ensure that the "force 1200 baud" jumper is not set inside the CX801.
- Set 9600 baud 7 bit /odd parity at the front panel.
- Switch the unit to remote mode.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Watchdog)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Program checksum)
internal.ss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sp	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.pe	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
automatic	CHOICE		EnumRange (MANUAL AUTOMATIC)
demTimeout	INTEGER		IntegerRange (2 .. 255)
modTimeout	FLOAT		DoubleRange (0.0 .. 9.9)
modemTiming	CHOICE		EnumRange (EXTERNAL INTERNAL)
monitoring	CHOICE		EnumRange (MOD/DEM MOD MOD-HW DEM DEM-HW MOD-HW/DEM MOD/DEM-HW MOD-HW/DEM-HW)
shadow	TEXT		StringRange
spare	CHOICE		EnumRange (1 2 3 4 5 6 7 8 NONE)
protect.1	CHOICE		EnumRange (OFF ON)
protect.2	CHOICE		EnumRange (OFF ON)
protect.3	CHOICE		EnumRange (OFF ON)
protect.4	CHOICE		EnumRange (OFF ON)
protect.5	CHOICE		EnumRange (OFF ON)
protect.6	CHOICE		EnumRange (OFF ON)
protect.7	CHOICE		EnumRange (OFF ON)
protect.8	CHOICE		EnumRange (OFF ON)
enable.1	CHOICE		EnumRange (OFF ON)
enable.2	CHOICE		EnumRange (OFF ON)
enable.3	CHOICE		EnumRange (OFF ON)
enable.4	CHOICE		EnumRange (OFF ON)
enable.5	CHOICE		EnumRange (OFF ON)
enable.6	CHOICE		EnumRange (OFF ON)
enable.7	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
enable.8	CHOICE		EnumRange (OFF ON)
enable.S	CHOICE		EnumRange (OFF ON)
modemFault.1	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.2	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.3	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.4	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.5	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.6	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.7	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.8	CHOICE	R/O	EnumRange R/O (OK FLT)
modemFault.S	CHOICE	R/O	EnumRange R/O (OK FLT)

9.130 Comstream-DAC700-PRE

This page describes the device driver and the device window for the Comstream DAC700 Digital Audio Codec. There are two variants of the driver available. The *Comstream-DAC700* driver works with both, the RS232 and the RS485 interface but does not support device presets. The *Comstream-DAC700-PRE* driver is limited for usage with the RS232 interface but in return implements the device preset functions.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The encoder settings
- --- The decoder settings
- --- The parameters concerning user data multiplexing
- --- Shows the actual relay I/O status and defines the relay contact sense / mapping.
- --- Lets you edit the device presets in the DAC700. This page is available with the *Comstream-DAC700-PRE* driver only.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DAC700.

parameter	***description***
address	If you are using the packet protocol / RS485 you must set this to the packet address set in the DAC700. Valid addresses are 1..31, 31 is the factory default. If you have multiple codecs connected to one RS485 bus, each device must have it's unique address. With the RS232 interface, the packet address doesn't care, you can leave this field empty.
faultMask	Defines the fault mask applied to the status bits in the DAC700 ("AM" command). See the Comstream manual for details.
Serial	Remember to set the line parameters for the serial interface connected to the DAC700. While the DAC700'a factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* . See below how to change this setting at the DAC700.

Communication protocol / Interface type

The DAC700 may be connected to the M&C computer / VLC via RS232 using the*Comstream-ASCII-Protocol* or by RS485 using the *Comstream-Packet-Protocol* . The *Comstream-DAC700-PRE* driver version requires a RS232 interface.

Installation

The DAC700 does not have a front panel control. In order to prepare the DAC700 for use with the sat-nms software, connect an ASCII terminal (or a PC running a terminal program) to the DAC700. If you have a 9 pin RS232 connector at your terminal, you need a straight through 9-pin cable (male/female) for this. The DAC700's factory line settings are 1200,O,7,1, first try this setting at your terminal. Now command the following settings:

command	***description***
RA 31	Sets the codecs packet address. You may choose any value between 1 and 31 as long as you set the same value at the *address* configuration parameter in the software. If you are operating multiple devices on one common interface, ensure that each device is set to a unique address. You may skip this if you intend to use the RS232 port for M&C.
RC 9600,O,7,1	Sets the receivers RS485 port to the recommended line settings. You may skip this if you intend to use the RS232 port for M&C.

command	*description*
PC 9600,O,7,1	Sets the receivers RS232 port to the recommended line settings. Your terminal may show strange characters after you entered this, because it does not understand the data sent by the DAC700 unless you change the line settings at the terminal, too. You may skip this if the DAC700 shall be controlled vie RS485.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
reset	TEXT	NOPRESET	StringRange
enc.dataRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
enc.channelID	INTEGER		IntegerRange (1 .. 255)
enc.networkID	INTEGER		IntegerRange (1 .. 255)
enc.mode	CHOICE		EnumRange (MONO DUAL MONO JOINT STEREO)
enc.bandwidth	CHOICE		EnumRange (BW-A BW-B BW-C BW-D)
enc.algorithm	CHOICE		EnumRange (Musicam ISO/MPEG)
enc.clockSource	CHOICE		EnumRange (INTERNAL EXT-ST EXT-RT AES/EBU)

name	*type*	*flags*	*range*
dec.dataRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
dec.protection	CHOICE		EnumRange (OFF ON)
dec.loopBack	CHOICE		EnumRange (OFF PCM-LOOP MPEG-LOOP)
dec.ibsSelect	CHOICE		EnumRange (NONE RELAY RELAY+DATA ALL)
dec.lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)
dec.audioMute	CHOICE		EnumRange (UNMUTED MUTED MUTED-L MUTED-R)
dec.testTone	CHOICE		EnumRange (OFF 1kHz LEFT 1kHz RIGHT 1kHz BOTH 9.6kHz LEFT 9.6kHz RIGHT 9.6kHz BOTH)
contactMap.1	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.2	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.3	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.4	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.5	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.6	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.7	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.8	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactSense.1	CHOICE		EnumRange (NRM INV)
contactSense.2	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
contactSense.3	CHOICE		EnumRange (NRM INV)
contactSense.4	CHOICE		EnumRange (NRM INV)
contactSense.5	CHOICE		EnumRange (NRM INV)
contactSense.6	CHOICE		EnumRange (NRM INV)
contactSense.7	CHOICE		EnumRange (NRM INV)
contactSense.8	CHOICE		EnumRange (NRM INV)
contact.1	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.2	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.3	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.4	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.5	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.6	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.7	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.8	CHOICE	R/O	EnumRange R/O (OFF ON)
relay.mask	INTEGER		IntegerRange (0 .. 65535)
relay.invert	INTEGER		IntegerRange (0 .. 65535)
relay.debounce	INTEGER		IntegerRange (20 .. 1000)
mux.relayInterval	INTEGER		IntegerRange (0 .. 10000)
mux.dataInterval	INTEGER		IntegerRange (0 .. 1000)
mux.dataTermination	INTEGER		IntegerRange (-128 .. 127)
mux.duplexMode	CHOICE		EnumRange (SIMPLEX DUPLEX)
ancillary.rate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
ancillary.parity	CHOICE		EnumRange (O E N)
ancillary.databits	CHOICE		EnumRange (8)
ancillary.stopbits	CHOICE		EnumRange (2)

name	*type*	*flags*	*range*
ancillary.blockLength	INTEGER		IntegerRange (30 .. 255)
data2.rate	CHOICE		EnumRange (1200 2400 4800 9600)
data2.parity	CHOICE		EnumRange (O E N)
data2.databits	CHOICE		EnumRange (7 8)
data2.stopbits	CHOICE		EnumRange (1 2)
data2.enable	CHOICE		EnumRange (OFF ON)
data1.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)
data1.parity	CHOICE		EnumRange (O E N)
data1.databits	CHOICE		EnumRange (7 8)
data1.stopbits	CHOICE		EnumRange (1 2)
preset.1.networkID	INTEGER		IntegerRange (1 .. 255)
preset.1.channelID	INTEGER		IntegerRange (1 .. 255)
preset.1.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.1.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.1.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.1.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.1.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.2.networkID	INTEGER		IntegerRange (1 .. 255)
preset.2.channelID	INTEGER		IntegerRange (1 .. 255)
preset.2.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.2.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)

name	*type*	*flags*	*range*
preset.2.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.2.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.2.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.3.networkID	INTEGER		IntegerRange (1 .. 255)
preset.3.channelID	INTEGER		IntegerRange (1 .. 255)
preset.3.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.3.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.3.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.3.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.3.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.4.networkID	INTEGER		IntegerRange (1 .. 255)
preset.4.channelID	INTEGER		IntegerRange (1 .. 255)
preset.4.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.4.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.4.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.4.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)

name	*type*	*flags*	*range*
preset.4.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.5.networkID	INTEGER		IntegerRange (1 .. 255)
preset.5.channelID	INTEGER		IntegerRange (1 .. 255)
preset.5.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.5.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.5.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.5.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.5.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.6.networkID	INTEGER		IntegerRange (1 .. 255)
preset.6.channelID	INTEGER		IntegerRange (1 .. 255)
preset.6.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.6.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.6.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.6.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.6.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.7.networkID	INTEGER		IntegerRange (1 .. 255)
preset.7.channelID	INTEGER		IntegerRange (1 .. 255)
preset.7.mode	CHOICE		EnumRange (MONO DUAL JOINT)

name	*type*	*flags*	*range*
preset.7.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.7.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.7.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.7.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.8.networkID	INTEGER		IntegerRange (1 .. 255)
preset.8.channelID	INTEGER		IntegerRange (1 .. 255)
preset.8.mode	CHOICE		EnumRange (MONO DUAL JOINT)
preset.8.clockSource	CHOICE		EnumRange (INT EXT-ST EXT-RT AES/EBU)
preset.8.agRate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
preset.8.txRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.8.rxRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
preset.mode	CHOICE		EnumRange (OFF SWITCH MANUAL)
preset.selection	CHOICE		EnumRange (1 2 3 4 5 6 7 8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tx PLL lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Rx PLL lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Decoder frame sync)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio-L overload)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio-R overload)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Network/channel ID)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.131 Comstream-DAC700

This page describes the device driver and the device window for the Comstream DAC700 Digital Audio Codec. There are two variants of the driver available. The *Comstream-DAC700* driver works with both, the RS232 and the RS485 interface but does not support device presets. The *Comstream-DAC700-PRE* driver is limited for usage with the RS232 interface but in return implements the device preset functions.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The encoder settings
- --- The decoder settings
- --- The parameters concerning user data multiplexing
- --- Shows the actual relay I/O status and defines the relay contact sense / mapping.
- --- Lets you edit the device presets in the DAC700. This page is available with the *Comstream-DAC700-PRE* driver only.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DAC700.

parameter	*description*
address	If you are using the packet protocol / RS485 you must set this to the packet address set in the DAC700. Valid addresses are 1..31, 31 is the factory default. If you have multiple codecs connected to one RS485 bus, each device must have it's unique address. With the RS232 interface, the packet address doesn't care, you can leave this field empty.

parameter	*description*
faultMask	Defines the fault mask applied to the status bits in the DAC700 ("AM" command). See the Comstream manual for details.
Serial	Remember to set the line parameters for the serial interface connected to the DAC700. While the DAC700'a factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* . See below how to change this setting at the DAC700.

Communication protocol / Interface type

The DAC700 may be connected to the M&C computer / VLC via RS232 using the*Comstream-ASCII-Protocol* or by RS485 using the *Comstream-Packet-Protocol* . The *Comstream-DAC700-PRE* driver version requires a RS232 interface.

Installation

The DAC700 does not have a front panel control. In order to prepare the DAC700 for use with the sat-nms software, connect an ASCII terminal (or a PC running a terminal program) to the DAC700. If you have a 9 pin RS232 connector at your terminal, you need a straight through 9-pin cable (male/female) for this. The DAC700's factory line settings are 1200,O,7,1, first try this setting at your terminal. Now command the following settings:

command	*description*
RA 31	Sets the codecs packet address. You may choose any value between 1 and 31 as long as you set the same value at the *address* configuration parameter in the software. If you are operating multiple devices on one common interface, ensure that each device is set to a unique address. You may skip this if you intend to use the RS232 port for M&C.
RC 9600,O,7,1	Sets the receivers RS485 port to the recommended line settings. You may skip this if you intend to use the RS232 port for M&C.
PC 9600,O,7,1	Sets the receivers RS232 port to the recommended line settings. Your terminal may show strange characters after you entered this, because it does not understand the data sent by the DAC700 unless you change the line settings at the terminal, too. You may skip this if the DAC700 shall be controlled vie RS485.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.version	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
enc.dataRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
enc.channelID	INTEGER		IntegerRange (1 .. 255)
enc.networkID	INTEGER		IntegerRange (1 .. 255)
enc.mode	CHOICE		EnumRange (MONO DUAL MONO JOINT STEREO)
enc.bandwidth	CHOICE		EnumRange (BW-A BW-B BW-C BW-D)
enc.algorithm	CHOICE		EnumRange (Musicam ISO/MPEG)
enc.clockSource	CHOICE		EnumRange (INTERNAL EXT-ST EXT-RT AES/EBU)
dec.dataRate	CHOICE		EnumRange (56000 64000 96000 112000 128000 192000 256000 384000)
dec.protection	CHOICE		EnumRange (OFF ON)
dec.loopBack	CHOICE		EnumRange (OFF PCM-LOOP MPEG-LOOP)
dec.ibsSelect	CHOICE		EnumRange (NONE RELAY RELAY+DATA ALL)
dec.lrToggle	CHOICE		EnumRange (NORMAL REVERSE LEFT RIGHT)
dec.audioMute	CHOICE		EnumRange (UNMUTED MUTED MUTED-L MUTED-R)

name	*type*	*flags*	*range*
dec.testTone	CHOICE		EnumRange (OFF 1kHz LEFT 1kHz RIGHT 1kHz BOTH 9.6kHz LEFT 9.6kHz RIGHT 9.6kHz BOTH)
contactMap.1	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.2	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.3	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.4	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.5	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.6	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.7	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactMap.8	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
contactSense.1	CHOICE		EnumRange (NRM INV)
contactSense.2	CHOICE		EnumRange (NRM INV)
contactSense.3	CHOICE		EnumRange (NRM INV)
contactSense.4	CHOICE		EnumRange (NRM INV)
contactSense.5	CHOICE		EnumRange (NRM INV)
contactSense.6	CHOICE		EnumRange (NRM INV)
contactSense.7	CHOICE		EnumRange (NRM INV)
contactSense.8	CHOICE		EnumRange (NRM INV)
contact.1	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.2	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.3	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
contact.4	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.5	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.6	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.7	CHOICE	R/O	EnumRange R/O (OFF ON)
contact.8	CHOICE	R/O	EnumRange R/O (OFF ON)
relay.mask	INTEGER		IntegerRange (0 .. 65535)
relay.invert	INTEGER		IntegerRange (0 .. 65535)
relay.debounce	INTEGER		IntegerRange (20 .. 1000)
mux.relayInterval	INTEGER		IntegerRange (0 .. 10000)
mux.dataInterval	INTEGER		IntegerRange (0 .. 1000)
mux.dataTermination	INTEGER		IntegerRange (-128 .. 127)
mux.duplexMode	CHOICE		EnumRange (SIMPLEX DUPLEX)
ancillary.rate	CHOICE		EnumRange (1200 2400 4800 9600 38400)
ancillary.parity	CHOICE		EnumRange (O E N)
ancillary.databits	CHOICE		EnumRange (8)
ancillary.stopbits	CHOICE		EnumRange (2)
ancillary.blockLength	INTEGER		IntegerRange (30 .. 255)
data2.rate	CHOICE		EnumRange (1200 2400 4800 9600)
data2.parity	CHOICE		EnumRange (O E N)
data2.databits	CHOICE		EnumRange (7 8)
data2.stopbits	CHOICE		EnumRange (1 2)
data2.enable	CHOICE		EnumRange (OFF ON)
data1.rate	CHOICE		EnumRange (300 1200 2400 4800 9600)
data1.parity	CHOICE		EnumRange (O E N)
data1.databits	CHOICE		EnumRange (7 8)

name	*type*	*flags*	*range*
data1.stopbits	CHOICE		EnumRange (1 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tx PLL lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Rx PLL lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Decoder frame sync)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio-L overload)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio-R overload)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Network/channel ID)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.132 Comstream-Demodulator

This page describes the device driver and the device window for a single Comstream Demodulator Card operated in a CM701 device frame. Beside the demodulator card, there may be other modulator, demodulator or modem cards installed in the CM701. In the software each card appears as a separate device.

The demodulator card device driver prepends the modem card's slot number to all commands and individually addresses the card in the modem frame this way. The modem does not support the "primary data rate" macros in this mode, hence the driver is limited to SCPC configurations.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	***group***	***description***
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The RX interface settings.
- --- The receive quality.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the modem card.

- *parameter --- description*
- address --- The modem frame RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address.
- demSlot --- You must set this parameter to the slot number (5..7) where the demodulator card you want to address is installed.
- ifcSlot --- You must set this parameter to the slot number (1..3) where the data interface used by this demodulator is installed.
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details. This parameter applies to all cards installed in the device frame.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask". This parameter applies to all cards installed in the device frame.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the modem. While a CM701's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM701 modems is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *The faults page only shows four fault flags. Where are all the detailed CM701 faults?* --- The CM701 in it's fault register shows the detailed flags for all modem cards ORed together. The demodulator card itself returns only a go/no-go flag. As it appears to be more useful to have the faults separated by card than by detail, the driver does not use

the main status register and displays the simple fault flag for each modem card instead.

- *The RX frequency field at the user interface contains no value. Why?*-- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.demSlot	CHOICE	SAVE SETUP	EnumRange (5: 6: 7:)
config.ifcSlot	CHOICE	SETUP	EnumRange (1: 2: 3:)
config.freqBand	CHOICE	SAVE SETUP	EnumRange (70 MHz 140 MHz L-Band)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
refClkSrc	CHOICE		EnumRange (INT)

name	*type*	*flags*	*range*
rx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 2340000)
rx.dem.symbolRate	INTEGER		IntegerRange (4800 .. 2340000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NO 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (0 .. 255)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Demodulator)
faults.02	ALARM	R/O	AlarmFlagRange R/O (M&C)
faults.03	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.133 Comstream-LoSpeed-Modem-SCPC

This page describes the device driver and the device window for the Comstream Lo-Speed Modem Card. With this driver you can address up to three modem cards in one CM701 frame as separate devices. To do this, simply configure multiple modems of this type at the same serial port and setup the configuration parameters for each particular modem / modem card as described later in this chapter.

The modem card device driver prepends the modem card's slot number to all commands and individually addresses the card in the modem frame this way. The modem does not support the "primary data rate" macros in this mode, hence the driver is limited to SCPC configurations.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	***group***	***description***
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).

driver name	***group***	***description***
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The TX interface settings.
- --- The RX interface settings.
- --- The receive quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the modem card.

- *parameter --- description*
- address --- The modem frame RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address.
- slot --- You must set this parameter to the slot number (5..7) where the modem card you want to address is installed.
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details. This parameter applies to all modem cards installed in the device frame.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask". This parameter applies to all modem cards installed in the device frame.
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the modem. While a CM701's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM601/CM701 modems is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *The faults page only shows four fault flags. Where are all the detailed CM701 faults?*--- The CM701 in it's fault register shows the detailed flags for all modem cards ORed together. The modem card itself returns only a go/no-go flag. As it appears to be more useful to have the faults separated by modem than by detail, the driver does not use the main status register and displays the simple fault flag for each modem card instead.
- *One of the IF frequency fields at the user interface contains no value. Why?*-- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.
- *What do the "set to xxx but reported yyy" messages in the event log mean?*--- The range of some parameters of the CM701 depend on other parameter settings. For instance, the Reed-Solomon scrambling type only can be selected, if the Reed-Solomon codec is present and enabled. If not, the CM701 reverts to the old setting an one of these messages appears in the event log. Another parameter commonly causing such messages is the doppler buffer size. The value always must be a power of two, the CM701 automatically uses the next possible value above if you enter some other numbers.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.slot	CHOICE	SAVE SETUP	EnumRange (5: 6: 7:)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)
config.acqBits	INTEGER	SETUP	IntegerRange (0 .. 0)
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
modemType	CHOICE		EnumRange (SCPC)
refClkSrc	CHOICE		EnumRange (INT)
loopBack	CHOICE		EnumRange (OFF NEAR FAR)
tx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 512000)

name	*type*	*flags*	*range*
tx.mod.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
rx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)

name	*type*	*flags*	*range*
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.buf.size	INTEGER		IntegerRange (64 .. 8192)
rx.buf.mode	CHOICE		EnumRange (OFF ON)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (0 .. 255)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 100)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.handshake	CHOICE	R/O	EnumRange R/O (IGNORE RESPOND)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Modem card)
faults.02	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.03	ALARM	R/O	AlarmFlagRange R/O (M&C)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.134 Comstream-Modem-Card-SCPC

This page describes the device driver and the device window for the Comstream 5MBit and 512kBit Modem Cards. With this driver you can address up to three modem cards in one CM701 frame as separate devices. To do this, simply configure multiple modems of this type at the same serial port and setup the configuration parameters for each particular modem / modem card as described later in this chapter.

The modem card device driver prepends the modem card's slot number to all commands and individually addresses the card in the modem frame this way. The modem does not support the "primary data rate" macros in this mode, hence the driver is limited to SCPC configurations.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The

modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	***group***	***description***
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The TX interface settings.
- --- The RX interface settings.
- --- The receive quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the modem card.

- *parameter --- description*
- address --- The modem frame RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address.
- slot --- You must set this parameter to the slot number (5..7) where the modem card you want to address is installed.
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details. This parameter applies to all modem cards installed in the device frame.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask". This parameter applies to all modem cards installed in the device frame.
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- vsatMode --- Turns the VSAT mode on/off
- demodLength --- The "DL" parameter. See the Comstream manual for details.
- Serial --- Remember to set the line parameters for the serial interface connected to the modem. While a CM701's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM601/CM701 modems is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *One of the IF frequency fields at the user interface contains no value. Why?*-- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.
- *What do the "set to xxx but reported yyy" messages in the event log mean?*-- The range of some parameters of the CM701 depend on other parameter settings. For instance, the Reed-Solomon scrambling type only can be selected, if the Reed-Solomon codec is present and enabled. If not, the CM701 reverts to the old setting an one of these messages appears in the event log. Another parameter commonly causing such messages is the doppler buffer size. The value always must be a power of two, the CM701 automatically uses the next possible value above if you enter some other numbers.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.type	CHOICE	SAVE SETUP	EnumRange (512 kBit 5 MBit)
config.slot	CHOICE	SAVE SETUP	EnumRange (5: 6: 7:)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)

name	*type*	*flags*	*range*
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
config.vsatMode	CHOICE	SETUP	EnumRange (SCPC VSAT)
config.demodLength	CHOICE	SETUP	EnumRange (SHORT LONG AUTO)
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
modemType	CHOICE		EnumRange (SCPC)
refClkSrc	CHOICE		EnumRange (INT)
loopBack	CHOICE		EnumRange (OFF IF NEAR FAR)
tx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)

name	*type*	*flags*	*range*
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.hardware	CHOICE		EnumRange (V.35 RS-530 RS-232 NONE)
tx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
rx.frequency	FLOAT		DoubleRange (52.000 .. 176.000)
rx.dem.dataRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.symbolRate	INTEGER		IntegerRange (4800 .. 512000)
rx.dem.type	CHOICE		EnumRange (QPSK BPSK)
rx.dem.fec	CHOICE		EnumRange (NONE 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
rx.dem.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
rx.dem.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
rx.dem.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
rx.dem.sweepMode	CHOICE		EnumRange (STOP NORMAL FAST)
rx.dem.sweepOffset	INTEGER		IntegerRange (-30000 .. 30000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 500000)
rx.buf.size	INTEGER		IntegerRange (64 .. 262144)
rx.buf.mode	CHOICE		EnumRange (OFF ON)
rx.meter.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 21.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 100)
rx.limit.ebno	FLOAT		DoubleRange (0.0 .. 21.0)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
rx.ifc.hardware	CHOICE	R/O	EnumRange R/O (V.35 RS-530 RS-232 NONE)
rx.ifc.handshake	CHOICE	R/O	EnumRange R/O (IGNORE RESPOND)
faults.01	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Other)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Tx power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (TxIF synth)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Tx bit timing)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Horizontal Tx)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Tx fifo full)
faults.09	ALARM	R/O	AlarmFlagRange R/O (NVRAM access)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RxIF synth)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Rx AGC range)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Bit time sync)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Carr track sync)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Decoder sync)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Acquisition)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Rx carr range)
faults.17	ALARM	R/O	AlarmFlagRange R/O (RxBT range)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Horizontal Rx)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Low Eb/No)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Rx buffer overflow)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Rx buffer underflow)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Buffer output clock)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (BERT sync)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Output buffer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (IO port match)
faults.26	ALARM	R/O	AlarmFlagRange R/O (M&C)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.135 Comstream-Modulator

This page describes the device driver and the device window for a single Comstream Modulator Card operated in a CM701 device frame. Beside the modulator card, there may be other modulator, demodulator or modem cards installed in the CM701. In the software each card appears as a separate device.

The modulator card device driver prepends the modem card's slot number to all commands and individually addresses the card in the modem frame this way. The modem does not support the "primary data rate" macros in this mode, hence the driver is limited to SCPC configurations.

Comstream CM701 device driver overview

The Comstream CM701 satellite modem may be operated in many different configurations. The modular architecture of this device permits to compose a modem from modulator, demodulator and option cards as well as to run up to three modem cards in one device frame.

With the sat-nms software, there is a family of device drivers which cover the most usual configurations for the CM701. Basically there are two groups of drivers: The first one covers applications which make use of the option cards like the framer/buffer card for IDR or the Reed-Solomon codec. This group is limited to single channel (one modulator, one demodulator) usage.

The other group covers multi-channel applications which run up to three modem, modulator oder demodulator cards in one device frame. These drivers are limited to plain SCPC applications, they do not support any option cards. You may put two modem cards and a modulator card in a CM701, these cards will be treated individual devices with the sat-nms software. Below there is a list of the available drivers for the CM701:

driver name	*group*	*description*

driver name	*group*	*description*
Comstream-CM701	SINGLE-CH	This is the CM701 device driver, covering a 'traditional' modem configuration of one modulator, one demodulator and optionally a frame/buffer card or a Reed-Solomon option card. The doppler buffer option card is supported, too.
Comstream-CM701-MC	SINGLE-CH	Like the Comstream-CM701 driver this one also supports option cards. It is used with configurations that use a 5MBit modem card instead of the modulator / demodulator cards.
Comstream-Modulator	MULTI-CH	This driver controls a single modulator card (together with an IO card).
Comstream-Demodulator	MULTI-CH	This driver controls a single demodulator card (together with an IO card).
Comstream-Modem-Card-SCPC	MULTI-CH	This is a driver for the recent version of the modem card (5Mbit or 512 kBit).
Comstream-LoSpeed-Modem-SCPC	MULTI-CH	This is a driver for the old 512 kBit model of the modem card. You may identify this card by the label "Low Speed Modem / Buffer".

Modem cards, modulator cards and demodulator cards may be mixed in one CM701 frame if they are operated with a driver marked as MULTI-CH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking. buffer.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the modem card.

- *parameter --- description*
- address --- The modem frame RS485 packet address. Valid addresses are 1..31, 31 is the factory default. If you have multiple devices connected to one RS485 bus, each device must have it's unique address.
- modSlot --- You must set this parameter to the slot number (5..7) where the modulator card you want to address is installed.
- ifcSlot --- You must set this parameter to the slot number (1..3) where the data interface used by this modulator is installed.
- faultMask --- The modem's fault mask (both, the STO and the FO mask). Common values are:
  - 4294967295 --- All faults enabled.
  - 1440583691 --- RX faults disabled (TX-ONLY mode)
  - 2863594484 --- TX faults disabled (RX-ONLY mode)

See the Comstream manual for mode details. This parameter applies to all modem cards installed in the device frame.

- faultLogMask --- The modem's fault log mask (FLO), coded in the same way as "faultMask". This parameter applies to all modem cards installed in the device frame.
- txAutoOn --- The carrier state when the modem gets powered on.
- txFaultOff --- Defines if the modems switches off the carrier on some faults conditions.
- Serial --- Remember to set the line parameters for the serial interface connected to the modem. While a CM701's factory preset setting is *1200,O,7,1* , the recommended line setting for the use with the sat-nms software is *9600,O,7,1* .

Communication protocol / Interface type

The recommended communication protocol for the CM601/CM701 modems is the *Comstream-Packet-Protocol* (The driver works with the *Comstream-ASCII-Protocol* , too, but there are several issues when using the ASCII protocol (unsolicited "ST/FL" reports, multi-line terminal commands). You are free to use either a RS232 or a RS485 interface to access the modem.

Commonly Asked Questions

The CM701 does not seamlessly fit into the M&C software concept at some points. Commonly asked questions arising from these issues are answered below.

- *The faults page only shows four fault flags. Where are all the detailed CM701 faults?*--- The CM701 in it's fault register shows the detailed flags for all modem cards ORed together. The demodulator card itself returns only a go/no-go flag. As it appears to be more useful to have the faults separated by card than by detail, the driver does not use the main status register and displays the simple fault flag for each modem card instead.
- *The TX frequency field at the user interface contains no value. Why?*--- The CM701 uses a flexible format for IF frequency values. It remembers the number of digits the last command contained and replies to all following requests with this number of digits. This may lead to a wrong reading if the modem is operated with the software the fist time. Once the frequency has been set by the software one time, the CM701 reports the correct frequency value and the software display this.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.modSlot	CHOICE	SAVE SETUP	EnumRange (5: 6: 7:)
config.ifcSlot	CHOICE	SETUP	EnumRange (1: 2: 3:)
config.freqBand	CHOICE	SAVE SETUP	EnumRange (70 MHz 140 MHz)
config.faultMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.faultLogMask	INTEGER	SETUP	IntegerRange (0 .. 0)
config.txAutoOn	CHOICE	SETUP	EnumRange (OFF LAST)
config.txFaultOff	CHOICE	SETUP	EnumRange (IGNORE TX-OFF)
info.id	TEXT	R/O	StringRange R/O
info.config	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
refClkSrc	CHOICE		EnumRange (INT)
tx.frequency	FLOAT		DoubleRange (52.000 .. 88.000)
tx.power	FLOAT		DoubleRange (-25.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.dataRate	INTEGER		IntegerRange (4800 .. 2340000)
tx.mod.symbolRate	INTEGER		IntegerRange (4800 .. 2340000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NO 1/2-SEQ 3/4-SEQ 1/2-VIT 3/4-VIT 7/8-VIT)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL LOOP)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF COMSTRM IDR)
tx.mod.filter	CHOICE		EnumRange (COMSTRM INTELSAT EUTELSAT)
tx.mod.differential	CHOICE		EnumRange (OFF STANDARD QPSK-UNC)
tx.ifc.clockPhase	CHOICE		EnumRange (TT-CLK ST-CLK)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.handshake	CHOICE		EnumRange (IGNORE RESPOND)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Modulator)
faults.02	ALARM	R/O	AlarmFlagRange R/O (M&C)
faults.03	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
internal.st	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.136 Comtech-CDD564

This page describes the device driver and the device window for the Comtech CDD564AL device.

It supports four RF SAT inputs and support one decoder for each physical input signal. The driver only supports the receive parameter settings of each channel. Parameters for transmitting the received and demodulated data over ethernet have to be manually set in the web-interface.

The driver was tested with the following sw releases:

- **1.5.6 (release 24.07.2019)**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the parameters for each of the four decoders that correspond to the RX1 to RX4 inputs. Lets the user set frequency, data-/symbolrate, spectrum, modulation, rolloff etc. Also displays the "locked" state for each demodulator.
- --- Contains the LNB control for each of the four RF inputs. Lets the user set the DC voltage to the LNBs and optionally enable a 10MHz multiplexed reference signal. Also displays the actual voltage and current.
- --- Displays the receive parameters for each channel such as signal level, eb/n0, frequency shift and BER.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- dem.1.rxAGC --- Set the automatic gain control of LNB 1 to "active" or "masked".
- dem.1.lnb --- Set the LNB 1 to "active" or "masked".

- dem.1.EbNo --- Set Eb/No for channel 1 to "active" or "masked".
- dem.2.rxAGC --- Set the automatic gain control of LNB 2 to "active" or "masked".
- dem.2.lnb --- Set the LNB 2 to "active" or "masked".
- dem.2.EbNo --- Set Eb/No for channel 2 to "active" or "masked".
- dem.3.rxAGC --- Set the automatic gain control of LNB 3 to "active" or "masked".
- dem.3.lnb --- Set the LNB 3 to "active" or "masked".
- dem.3.EbNo --- Set Eb/No for channel 3 to "active" or "masked".
- dem.4.rxAGC --- Set the automatic gain control of LNB 4 to "active" or "masked".
- dem.4.lnb --- Set the LNB 4 to "active" or "masked".
- dem.4.EbNo --- Set Eb/No for channel 4 to "active" or "masked".

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.dem.1.rxAGC	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.1.lnb	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.1.EbNo	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.2.rxAGC	CHOICE	SAVE SETUP	EnumRange (active masked)

name	*type*	*flags*	*range*
config.dem.2.lnb	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.2.EbNo	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.3.rxAGC	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.3.lnb	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.3.EbNo	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.4.rxAGC	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.4.lnb	CHOICE	SAVE SETUP	EnumRange (active masked)
config.dem.4.EbNo	CHOICE	SAVE SETUP	EnumRange (active masked)
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.appVersion	TEXT	R/O	StringRange R/O
info.bootinfo	TEXT	R/O	StringRange R/O
info.appimage1info	TEXT	R/O	StringRange R/O
info.appimage2info	TEXT	R/O	StringRange R/O
info.dataRate	TEXT	R/O	StringRange R/O
info.8psk8qam	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.16qam	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.turbo	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.VersaFec	CHOICE	R/O	EnumRange R/O (installed uninstalled)

name	*type*	*flags*	*range*
info.rolloff	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.rx3xdesDecryption	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.rxHeaderCompression	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.rxPayloadCompression	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.vipersat	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.vipersatFileStreamer	CHOICE	R/O	EnumRange R/O (installed uninstalled)
info.cpuLoad	INTEGER	R/O	IntegerRange R/O (0 .. 100)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmHelper.9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (+5V Power Supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (+12V Power Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (-5V Power Supply)
faults.04	ALARM	R/O	AlarmFlagRange R/O (-12V Power Supply)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (RX 1st LO synth Block)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RX 2st LO synth Block)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Dem1 Unlocked)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Dem1 AGC out of Range)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Dem1 Buffer Underflow)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Dem1 Buffer Overflow)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Dem1 Eb/No)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Dem1 LNB)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Dem1 AIS detect on data)
faults.14	ALARM	R/O	AlarmFlagRange R/O (LNB1 Current)
faults.15	ALARM	R/O	AlarmFlagRange R/O (LNB1 Voltage)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Dem2 Unlocked)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Dem2 AGC out of Range)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Dem2 Buffer Underflow)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Dem2 Buffer Overflow)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Dem2 Eb/No)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Dem2 LNB)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Dem2 AIS detect on data)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (LNB2 Current)
faults.24	ALARM	R/O	AlarmFlagRange R/O (LNB2 Voltage)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Dem3 Unlocked)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Dem3 AGC out of Range)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Dem3 Buffer Underflow)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Dem3 Buffer Overflow)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Dem3 Eb/No)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Dem3 LNB)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Dem3 AIS detect on data)
faults.32	ALARM	R/O	AlarmFlagRange R/O (LNB3 Current)
faults.33	ALARM	R/O	AlarmFlagRange R/O (LNB3 Voltage)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Dem4 Unlocked)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Dem4 AGC out of Range)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Dem4 Buffer Underflow)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Dem4 Buffer Overflow)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Dem4 Eb/No)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Dem4 LNB)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Dem4 AIS detect on data)

name	*type*	*flags*	*range*
faults.41	ALARM	R/O	AlarmFlagRange R/O (LNB4 Current)
faults.42	ALARM	R/O	AlarmFlagRange R/O (LNB4 Voltage)
rx.dem.1.frequency	FLOAT		DoubleRange (950.0000 .. 2250.0000)
rx.dem.1.dataRate	FLOAT		DoubleRange (16.000 .. 10239.005)
rx.dem.1.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.dem.1.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.1.descrambler	CHOICE		EnumRange (Off On(default) IESS-315)
rx.dem.1.sweepRange	INTEGER		IntegerRange (0 .. 32)
rx.dem.1.ebnoAlarm	FLOAT		DoubleRange (0.1 .. 16.0)
rx.dem.1.fecType	CHOICE		EnumRange (Uncoded TPC)
rx.dem.1.demodType	CHOICE		EnumRange (BPSK QPSK 8PSK 8-QAM 16-QAM)
rx.dem.1.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
rx.dem.1.dataInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.1.clockInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.1.rolloff	CHOICE		EnumRange (0.35 0.25 0.20 0.15 0.10 0.05)
rx.dem.1.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.dem.1.lockstatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.2.frequency	FLOAT		DoubleRange (950.0000 .. 2250.0000)
rx.dem.2.dataRate	FLOAT		DoubleRange (16.000 .. 10239.005)
rx.dem.2.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
rx.dem.2.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.2.descrambler	CHOICE		EnumRange (Off On(default) IESS-315)
rx.dem.2.sweepRange	INTEGER		IntegerRange (0 .. 32)
rx.dem.2.ebnoAlarm	FLOAT		DoubleRange (0.1 .. 16.0)
rx.dem.2.fecType	CHOICE		EnumRange (Uncoded TPC)
rx.dem.2.demodType	CHOICE		EnumRange (BPSK QPSK 8PSK 8-QAM 16-QAM)
rx.dem.2.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
rx.dem.2.dataInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.2.clockInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.2.rolloff	CHOICE		EnumRange (0.35 0.25 0.20 0.15 0.10 0.05)
rx.dem.2.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.dem.2.lockstatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.3.frequency	FLOAT		DoubleRange (950.0000 .. 2250.0000)
rx.dem.3.dataRate	FLOAT		DoubleRange (16.000 .. 10239.005)
rx.dem.3.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.dem.3.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.3.descrambler	CHOICE		EnumRange (Off On(default) IESS-315)
rx.dem.3.sweepRange	INTEGER		IntegerRange (0 .. 32)
rx.dem.3.ebnoAlarm	FLOAT		DoubleRange (0.1 .. 16.0)
rx.dem.3.fecType	CHOICE		EnumRange (Uncoded TPC)
rx.dem.3.demodType	CHOICE		EnumRange (BPSK QPSK 8PSK 8-QAM 16-QAM)

name	*type*	*flags*	*range*
rx.dem.3.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
rx.dem.3.dataInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.3.clockInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.3.rolloff	CHOICE		EnumRange (0.35 0.25 0.20 0.15 0.10 0.05)
rx.dem.3.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.dem.3.lockstatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.4.frequency	FLOAT		DoubleRange (950.0000 .. 2250.0000)
rx.dem.4.dataRate	FLOAT		DoubleRange (16.000 .. 10239.005)
rx.dem.4.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.dem.4.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.4.descrambler	CHOICE		EnumRange (Off On(default) IESS-315)
rx.dem.4.sweepRange	INTEGER		IntegerRange (0 .. 32)
rx.dem.4.ebnoAlarm	FLOAT		DoubleRange (0.1 .. 16.0)
rx.dem.4.fecType	CHOICE		EnumRange (Uncoded TPC)
rx.dem.4.demodType	CHOICE		EnumRange (BPSK QPSK 8PSK 8-QAM 16-QAM)
rx.dem.4.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 1/1 21/44 5/16 0.95)
rx.dem.4.dataInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.4.clockInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.4.rolloff	CHOICE		EnumRange (0.35 0.25 0.20 0.15 0.10 0.05)
rx.dem.4.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.dem.4.lockstatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meter.activeInputFlows	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meter.activeOutputFlows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.1.signalLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.1.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.dem.1.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
rx.dem.1.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.2.signalLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.2.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.dem.2.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
rx.dem.2.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.3.signalLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.3.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.dem.3.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
rx.dem.3.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.4.signalLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.4.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.dem.4.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
rx.dem.4.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.1.supply	CHOICE		EnumRange (off 13V 18V)
rx.lnb.1.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.1.10mhzReference	CHOICE		EnumRange (off on)
rx.lnb.1.currAlarmLow	INTEGER		IntegerRange (10 .. 400)

name	*type*	*flags*	*range*
rx.lnb.1.currAlarmHigh	INTEGER		IntegerRange (50 .. 600)
rx.lnb.1.loFreq	INTEGER		IntegerRange (0 .. 0)
rx.lnb.1.loMix	CHOICE		EnumRange (lowSide highSide)
rx.lnb.1.rfFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.lnb.2.supply	CHOICE		EnumRange (off 13V 18V)
rx.lnb.2.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.2.10mhzReference	CHOICE		EnumRange (off on)
rx.lnb.2.currAlarmLow	INTEGER		IntegerRange (10 .. 400)
rx.lnb.2.currAlarmHigh	INTEGER		IntegerRange (50 .. 600)
rx.lnb.2.loFreq	INTEGER		IntegerRange (0 .. 0)
rx.lnb.2.loMix	CHOICE		EnumRange (lowSide highSide)
rx.lnb.2.rfFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.lnb.3.supply	CHOICE		EnumRange (off 13V 18V)
rx.lnb.3.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.3.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.3.10mhzReference	CHOICE		EnumRange (off on)
rx.lnb.3.currAlarmLow	INTEGER		IntegerRange (10 .. 400)
rx.lnb.3.currAlarmHigh	INTEGER		IntegerRange (50 .. 600)
rx.lnb.3.loFreq	INTEGER		IntegerRange (0 .. 0)
rx.lnb.3.loMix	CHOICE		EnumRange (lowSide highSide)
rx.lnb.3.rfFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
rx.lnb.4.supply	CHOICE		EnumRange (off 13V 18V)
rx.lnb.4.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.lnb.4.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
rx.lnb.4.10mhzReference	CHOICE		EnumRange (off on)
rx.lnb.4.currAlarmLow	INTEGER		IntegerRange (10 .. 400)
rx.lnb.4.currAlarmHigh	INTEGER		IntegerRange (50 .. 600)
rx.lnb.4.loFreq	INTEGER		IntegerRange (0 .. 0)
rx.lnb.4.loMix	CHOICE		EnumRange (lowSide highSide)
rx.lnb.4.rfFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)

9.137 Comtech-CDM570

This page describes the device driver and the device window for the Comtech CDM-570 and CDM-570L Satellite Modem. The drivers detects automatically model (570 or 570L) and adjusts the parameter ranges of frequency etc.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The Automatic Uplink Power Control settings.
- --- The data interface settings.
- --- The receive quality.
- --- LNB and BUC meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*

parameter	***description***
address	The modem's packet address. For RS232 use 0000. When controlling a CDM-570AL through an IP connection, then the address also contains the user name ans password for login (addr:name:password). Using "0000:comtech:comtech" will work with units using the factory settings.
alarmMask	Sets the alarm mask conditions.
edmacMode	Embedded Distant-end Monitor And Control (EDMAC) Framing Mode
edmacAddress	Sets the range of addresses of distant-end units, which the master will forward messages for. (EDMAC)
circuitID	sets the user-defined Circuit-ID string, with a fixed! length of 40 characters
lnb.power	Enable the supply voltage for an LNB
lnb.enableRef	Enable the reference signal for the LNB
low/highLimits	Defines the limits for the over/under current alarm
buc.power	Enable the supply voltage for an BUC
buc.enableRef	Enable the 10 MHz reference for the LNB
low/highLimits	Defines the limits for the over/under current alarm
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* . You have to set the same value at the device's front panel.

Remote interface

The CDM-570 provides a RS232 on the connector P4B at the rear panel labeled*Remote Control* . You can configure this interface on the front panel in the menu *Configuration | Remote* . Select *Remote Control* as operation mode, set interface type to*RS-232* (with the fixed address 0, the baudrate to 19k2, and configure the interface to *8 databits, none parity, 1 stopbit*.

To connect the modem to the sat-nms system use a 1:1 cable with the following pin assignment

- --- CDM-600VLC
- RX --- 22
- TX --- 33
- GND --- 55

The CDM-570AL version of the modem may be controlled through a Telnet/IP connection. For this case set the interface in the device configuration to "aaa.bbb.ccc.ddd:7023", where

aaa.bbb.ccc.ddd is the IP address of the modem. Select the "ComtechTelnet" communication protocol instead of "Comtech" in this case an set the "address" configuration parameter to "000:comtech:comtech".

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.alarmMask	TEXT	SETUP	StringRange
config.edmacMode	CHOICE	SETUP	EnumRange (Off Master Slave)
config.edmacAddress	INTEGER	SETUP	IntegerRange (0 .. 9990)
config.warmupDelay	CHOICE	SETUP	EnumRange (OFF ON)
config.rts	CHOICE	SETUP	EnumRange (loop/no action loop/tx op ignore 1:n system)
config.circuitID	TEXT	SETUP	StringRange
config.lnb.power	CHOICE	SETUP	EnumRange DIS (OFF 13V 18V 24V)
config.lnb.enableRef	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.lnb.lowCurrentLimit	INTEGER	SETUP	IntegerRange DIS (10 .. 400)
config.lnb.HighCurrentLimit	INTEGER	SETUP	IntegerRange DIS (50 .. 600)
config.buc.power	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.buc.10MHzRef	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.buc.lowCurrentLimit	INTEGER	SETUP	IntegerRange DIS (0 .. 3000)
config.buc.highCurrentLimit	INTEGER	SETUP	IntegerRange DIS (500 .. 4000)

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.activeImage	TEXT	R/O	StringRange R/O
info.turbo	TEXT	R/O	StringRange R/O
info.dataRate	TEXT	R/O	StringRange R/O
info.modulation	TEXT	R/O	StringRange R/O
info.rsCodec	TEXT	R/O	StringRange R/O
info.ipModule	TEXT	R/O	StringRange R/O
info.redundancy	TEXT	R/O	StringRange R/O
info.buc.option	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
refClkFreq	CHOICE		EnumRange (internal ext. 1MHz ext. 2MHz ext. 5MHz ext. 10MHz ext. 20MHz)
loopBack	CHOICE		EnumRange (OFF Tx CW Tx Alternating IF RF Digital I/O)
tx.frequency	FLOAT		DoubleRange (50.0000 .. 180.0000)
tx.power	FLOAT		DoubleRange (-40.0 .. -0.0)
tx.on	CHOICE		EnumRange (OFF ON RTI OFF by HW OFF by BUC)
tx.mod.dataRate	FLOAT		DoubleRange (4.800 .. 5000.000)
tx.mod.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
tx.mod.fec	CHOICE		EnumRange (None w/ Diff.On Viterbi Viterbi+RS TCM+RS Turbo TPC/LDPC)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8-PSK 16-QAM)
tx.mod.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
tx.mod.clockSource	CHOICE		EnumRange (Internal Tx Terrestrial Loop-Timed)
tx.mod.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
tx.mod.dataInvert	CHOICE		EnumRange (Normal Inverted)
tx.mod.scrambling	CHOICE		EnumRange (Off Default IESS-315)
tx.ifc.overheadType	CHOICE		EnumRange (Unframed EDMAC EDMAC-2)
tx.ifc.hardware	CHOICE		EnumRange (EIA-422/EIA-530 V.35 DCE EIA-232 (sync) G.703 T1 AMI G.703 T1 B8ZS G.703 E1 Unbal AMI G.703 Unbal HDB3 G.703 Bal AMI G7.03 Bal HDB3 IP)
rx.frequency	FLOAT		DoubleRange (50.0000 .. 180.0000)
rx.dem.dataRate	FLOAT		DoubleRange (4.800 .. 20000.000)
rx.dem.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.dem.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8-PSK 16-QAM)
rx.dem.fec	CHOICE		EnumRange (None w/ Diff.On Viterbi Viterbi+RS TCM+RS Turbo TPC/LDPC)
rx.dem.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
rx.dem.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.dataInvert	CHOICE		EnumRange (Normal Inverted)

name	*type*	*flags*	*range*
rx.dem.scrambling	CHOICE		EnumRange (Off Default IESS-315)
rx.dem.sweepOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.dem.sweepRange	INTEGER		IntegerRange (1 .. 32)
rx.buf.size	CHOICE		EnumRange (disabled 1024bit 2048bit 4096bit 8192bit 16384bit 32768bit)
rx.buf.recenter	TEXT	NOPRESET	StringRange
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.limit.ebno	FLOAT		DoubleRange (0.1 .. 16.0)
aupc.enable	CHOICE		EnumRange (ON OFF)
aupc.localAction	CHOICE		EnumRange (ON OFF)
aupc.remoteAction	CHOICE		EnumRange (nominal power max power)
aupc.target	FLOAT		DoubleRange (0.0 .. 9.9)
aupc.maxIncrease	FLOAT		DoubleRange (0 .. 9)
aupc.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
lnb.meter.current	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
lnb.meter.voltage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
buc.tx.on	CHOICE		EnumRange DIS (OFF ON)
buc.meter.current	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
buc.meter.voltage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply -5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply +18V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power supply -12V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Tx Synthesizer lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Rx1: Synthesizer lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Rx2: Synthesizer lock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ref PLL lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (IP Module)
faults.12	ALARM	R/O	AlarmFlagRange R/O (EEProm Checksum)
faults.13	ALARM	R/O	AlarmFlagRange R/O (No clock terrestrial int.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Tx FIFO slip)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Loss ext. reference)
faults.16	ALARM	R/O	AlarmFlagRange R/O (AUPC up.limit reached)
faults.17	ALARM	R/O	AlarmFlagRange R/O (TX: AIS detected)
faults.18	ALARM	R/O	AlarmFlagRange R/O (G.703 Bipolar violation)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Demod unlocked)
faults.20	ALARM	R/O	AlarmFlagRange R/O (AGC out of range)
faults.21	ALARM	R/O	AlarmFlagRange R/O (RS frame sync)
faults.22	ALARM	R/O	AlarmFlagRange R/O (EDMAC frame sync)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Buffer underflow)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Buffer overflow)
faults.25	ALARM	R/O	AlarmFlagRange R/O (eb/no)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX: AIS detected)
faults.27	ALARM	R/O	AlarmFlagRange R/O (BUC PLL)
faults.28	ALARM	R/O	AlarmFlagRange R/O (BUC current)
faults.29	ALARM	R/O	AlarmFlagRange R/O (BUC voltage)
faults.30	ALARM	R/O	AlarmFlagRange R/O (LNB current)
faults.31	ALARM	R/O	AlarmFlagRange R/O (LNB voltage)
faults.32	ALARM	R/O	AlarmFlagRange R/O (BUC temperatur)
faults.33	ALARM	R/O	AlarmFlagRange R/O (BUC checksum)
internal.a	HEX	R/O	HexRange R/O (0 .. 0)
internal.b	HEX	R/O	HexRange R/O (0 .. 0)
internal.c	HEX	R/O	HexRange R/O (0 .. 0)
internal.d	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.esc	TEXT	R/O	StringRange R/O
internal.text	TEXT	R/O	StringRange R/O

9.138 Comtech-CDM600

This page describes the device driver and the device window for the Comtech CDM-600 Satellite Modem. The driver actually supports the following modem operation modes:

- SCPC
- IDR overhead
- IBS overhead
- ASYNC overhead
- Drop/Insert
- AUPC

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The Reed-Solomon settings.
- --- The IDR/IBS mode settings.
- --- The ASYNC overhead settings.
- --- The Automatic Uplink Power Control settings.
- --- The data interface settings.
- --- The Drop&Insert settings.
- --- The receive quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*

parameter	***description***
address	The modem's RS485 packet address. Valid addresses are 1..255, 1 is the factory default. You must set the same address than set at the modem's front panel
alarmMask	Sets the alarm mask conditions.
edmacMode	Embedded Distant-end Monitor And Control (EDMAC) Framing Mode
edmacAddress	Sets the range of addresses of distant-end units, which the master will forward messages for. (EDMAC)
ODUCommunication	Enables/disables the communications via the FSK link with a Comtech EF Data Transceiver (ODU)
esc	Enables/disables the IBS high-rate ESC.
circuitID	sets the user-defined Circuit-ID string, with a fixed! length of 40 characters
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* . You have to set the same value at the device's front panel.

Remote interface

The CDM-600 provides a RS232 on the connector P4B at the rear panel labeled*Remote Control* . You can configure this interface on the front panel in the menu *Configuration | Remote* . Select *Remote Control* as operation mode, set interface type to*RS-232* (with the fixed address 0, the baudrate to 19k2, and configure the interface to *8 databits, none parity,1 stopbit* To connect the modem to the sat-nms system use a 1:1 cable with the following pin assignment

- --- CDM-600VLC
- RX --- 23
- TX --- 23
- GND --- 55

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.alarmMask	TEXT	SETUP	StringRange
config.edmacMode	CHOICE	SETUP	EnumRange (Off Master Slave)
config.edmacAddress	INTEGER	SETUP	IntegerRange (0 .. 9990)
config.esc	CHOICE	SETUP	EnumRange (OFF ON)
config.circuitID	TEXT	SETUP	StringRange
config.freqMode	CHOICE	SAVE SETUP	EnumRange (IF RF)
config.lnb.power	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.lnb.enableRef	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.lnb.voltage	CHOICE	SETUP	EnumRange DIS (13V 18V 24V)
config.lnb.lowCurrentLimit	INTEGER	SETUP	IntegerRange DIS (0 .. 500)
config.lnb.highCurrentLimit	INTEGER	SETUP	IntegerRange DIS (0 .. 500)
config.lnb.loFreq	INTEGER	SETUP	IntegerRange DIS (0 .. 65000)
config.lnb.loMix	CHOICE	SETUP	EnumRange DIS (HIGH- LOW+)
config.ODUCommunication	CHOICE	SETUP	EnumRange (OFF ON)
config.buc.address	INTEGER	SETUP	IntegerRange DIS (0 .. 15)
config.buc.power	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.buc.10MHzRef	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.buc.lowCurrentLimit	INTEGER	SETUP	IntegerRange DIS (0 .. 4000)
config.buc.highCurrentLimit	INTEGER	SETUP	IntegerRange DIS (0 .. 4000)

name	*type*	*flags*	*range*
config.buc.loFreq	INTEGER	SETUP	IntegerRange DIS (0 .. 65000)
config.buc.loMix	CHOICE	SETUP	EnumRange DIS (HIGH- LOW+)
config.buc.txPowerupDelay	TEXT	SETUP	StringRange DIS
lnb.meter.current	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
buc.tx.on	CHOICE		EnumRange DIS (OFF ON)
buc.meter.power	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
buc.meter.temperature	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
buc.meter.current	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
buc.meter.voltage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.turbo	TEXT	R/O	StringRange R/O
info.dataRate	TEXT	R/O	StringRange R/O
info.modulation	TEXT	R/O	StringRange R/O
info.framing	TEXT	R/O	StringRange R/O
info.dropInsert	TEXT	R/O	StringRange R/O
info.specialOption	TEXT	R/O	StringRange R/O
info.redundancy	TEXT	R/O	StringRange R/O
refClkSrc	CHOICE		EnumRange (Unbalanced Balanced)
refClkFreq	FLOAT		DoubleRange (2.400 .. 20000.000)
loopBack	CHOICE		EnumRange (OFF Tx CW Tx Alternating IF RF Digital I/O)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.frequency	FLOAT		DoubleRange (52.0000 .. 176.0000)

name	*type*	*flags*	*range*
tx.power	FLOAT		DoubleRange (-20.0 .. -0.0)
tx.on	CHOICE		EnumRange (OFF ON RTI OFF by HW)
tx.mod.dataRate	FLOAT		DoubleRange (4.800 .. 20000.000)
tx.mod.fec	CHOICE		EnumRange (Diff.On Viterbi Viterbi+RS Sequential Seq+RS TCM TCM+RS Turbo Diff.Off)
tx.mod.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8-PSK 16-QAM)
tx.mod.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
tx.mod.clockSource	CHOICE		EnumRange (Internal Tx Terrestrial Rx Loop External)
tx.mod.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
tx.mod.dataInvert	CHOICE		EnumRange (Normal Inverted)
tx.mod.scrambling	CHOICE		EnumRange (Off Normal Special)
tx.rs.encoding	CHOICE		EnumRange (Normal IESS-310 EF-Data IBS)
tx.ifc.overheadType	CHOICE		EnumRange (Unframed IBS IDR DROP EDMAC)
tx.ifc.framingMode	CHOICE		EnumRange (AMI B8ZS B6ZS HDB3)
tx.ifc.hardware	CHOICE		EnumRange (RS422 V.35 RS232 Bal. G.703 Unb. G.703 Audio LVDS)
tx.idr.overheadType	CHOICE		EnumRange (Data Audio)
tx.idr.bwAlarm.1	CHOICE		EnumRange (Off Intern Extern)
tx.idr.bwAlarm.2	CHOICE		EnumRange (Off Intern Extern)
tx.idr.bwAlarm.3	CHOICE		EnumRange (Off Intern Extern)
tx.idr.bwAlarm.4	CHOICE		EnumRange (Off Intern Extern)

name	*type*	*flags*	*range*
tx.idr.audioLevel.1	CHOICE		EnumRange (-6 -4 -2 +0 +2 +4 +6 +8)
tx.idr.audioLevel.2	CHOICE		EnumRange (-6 -4 -2 +0 +2 +4 +6 +8)
rx.frequency	FLOAT		DoubleRange (52.0000 .. 176.0000)
rx.dem.dataRate	FLOAT		DoubleRange (4.800 .. 20000.000)
rx.dem.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8-PSK 16-QAM)
rx.dem.fec	CHOICE		EnumRange (Diff.On Viterbi Viterbi+RS Sequential Seq+RS TCM TCM+RS Turbo Diff.Off)
rx.dem.fecRate	CHOICE		EnumRange (1/2 3/4 7/8 2/3 1/1 21/44 5/16 0.95)
rx.dem.clockSource	CHOICE		EnumRange (Internal Tx Terrestrial Rx Loop External)
rx.dem.spectrumInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.dataInvert	CHOICE		EnumRange (Normal Inverted)
rx.dem.scrambling	CHOICE		EnumRange (Off Normal Special)
rx.dem.sweepOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.sweepRange	INTEGER		IntegerRange (1 .. 32)
rx.buf.size	INTEGER		IntegerRange (16 .. 32768)
rx.buf.recenter	TEXT	NOPRESET	StringRange
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.limit.ebno	FLOAT		DoubleRange (0.1 .. 16.0)

name	*type*	*flags*	*range*
rx.rs.decoding	CHOICE		EnumRange (Normal IESS-310 EF-Data IBS)
rx.ifc.overheadType	CHOICE		EnumRange (Unframed IBS IDR DROP EDMAC)
rx.ifc.framingMode	CHOICE		EnumRange (AMI B8ZS B6ZS HDB3)
rx.ifc.hardware	CHOICE		EnumRange (RS422 V.35 RS232 Bal. G.703 Unb. G.703 Audio LVDS)
rx.idr.overheadType	CHOICE		EnumRange (Data Audio)
rx.idr.bwAlarm.1	CHOICE		EnumRange (Off Extern)
rx.idr.bwAlarm.2	CHOICE		EnumRange (Off Extern)
rx.idr.bwAlarm.3	CHOICE		EnumRange (Off Extern)
rx.idr.bwAlarm.4	CHOICE		EnumRange (Off Extern)
rx.idr.audioLevel.1	CHOICE		EnumRange (-6 -4 -2 +0 +2 +4 +6 +8)
rx.idr.audioLevel.2	CHOICE		EnumRange (-6 -4 -2 +0 +2 +4 +6 +8)
aupc.enable	CHOICE		EnumRange (ON OFF)
aupc.localAction	CHOICE		EnumRange (ON OFF)
aupc.remoteAction	CHOICE		EnumRange (nominal power max power)
aupc.target	FLOAT		DoubleRange (0.0 .. 9.9)
aupc.maxIncrease	FLOAT		DoubleRange (0 .. 9)
aupc.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
aupc.powerInc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.frm.dropType	CHOICE		EnumRange (T1-D4 T1-ESF E1-CCS E1-CAS)
rx.frm.insertType	CHOICE		EnumRange (T1-D4 T1-ESF E1-CCS E1-CAS)

name	*type*	*flags*	*range*
rx.frm.loop	CHOICE		EnumRange (ON OFF)
tx.frm.dropSlots1	TEXT		StringRange
tx.frm.dropSlots2	TEXT		StringRange
tx.frm.dropSlots3	TEXT		StringRange
tx.frm.dropSlots4	TEXT		StringRange
tx.frm.dropSlots5	TEXT		StringRange
tx.frm.dropSlots6	TEXT		StringRange
rx.frm.insertSlots1	TEXT		StringRange
rx.frm.insertSlots2	TEXT		StringRange
rx.frm.insertSlots3	TEXT		StringRange
rx.frm.insertSlots4	TEXT		StringRange
rx.frm.insertSlots5	TEXT		StringRange
rx.frm.insertSlots6	TEXT		StringRange
tx.frm.terrestrialAlarm	CHOICE	SETUP	EnumRange (OFF ON)
rx.frm.terrestrialAlarm	CHOICE	SETUP	EnumRange (OFF ON)
tx.asy.baudrate	CHOICE		EnumRange (1200 2400 4800 9600 14400 19200 28800 38400)
tx.asy.parity	CHOICE		EnumRange (None Even Odd)
tx.asy.dataBits	INTEGER	R/O	IntegerRange R/O (0 .. 8)
tx.asy.stopBits	INTEGER	R/O	IntegerRange R/O (0 .. 8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply -5V)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply +18V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power supply -12V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RAM load fail)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Tx Synthesizer lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Rx Synthesizer)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Power cal checksum)
faults.11	ALARM	R/O	AlarmFlagRange R/O (FPGA main chain load)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Turbo FPGA load fail)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Modem card FPGA load)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Mux FPGA load)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Demux FPGA load)
faults.16	ALARM	R/O	AlarmFlagRange R/O (RX synthesizer 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (No lock terrestrial int.)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Tx FIFO slip)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Tx: AIS incoming data)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (AUPC up.limit reached)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Ref PLL)
faults.22	ALARM	R/O	AlarmFlagRange R/O (BUC current)
faults.23	ALARM	R/O	AlarmFlagRange R/O (BUC voltage)
faults.24	ALARM	R/O	AlarmFlagRange R/O (BUC no comms or chksum)
faults.25	ALARM	R/O	AlarmFlagRange R/O (BUC PLL)
faults.26	ALARM	R/O	AlarmFlagRange R/O (BUC temperature)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Demodulator unlocked)
faults.28	ALARM	R/O	AlarmFlagRange R/O (AGC out of range)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Demux)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Buffer slip)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Rx:AIS incoming data)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Eb/No alarm)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Buffer Clock activity)
faults.34	ALARM	R/O	AlarmFlagRange R/O (LNB current)
faults.35	ALARM	R/O	AlarmFlagRange R/O (LNB voltage)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Loss of Tx frame)
faults.37	ALARM	R/O	AlarmFlagRange R/O (BER)

name	*type*	*flags*	*range*
faults.38	ALARM	R/O	AlarmFlagRange R/O (Loss Tx multiframe)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Tx signaling AIS)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Tx remote Alarm)
faults.41	ALARM	R/O	AlarmFlagRange R/O (IBS satellite alarm)
faults.42	ALARM	R/O	AlarmFlagRange R/O (IDR Rx BWA1)
faults.43	ALARM	R/O	AlarmFlagRange R/O (IDR Rx BWA2)
faults.44	ALARM	R/O	AlarmFlagRange R/O (IDR Rx BWA3)
faults.45	ALARM	R/O	AlarmFlagRange R/O (IDR Rx BWA4)
faults.46	ALARM	R/O	AlarmFlagRange R/O (IDR Tx BWA1)
faults.47	ALARM	R/O	AlarmFlagRange R/O (IDR Tx BWA2)
faults.48	ALARM	R/O	AlarmFlagRange R/O (IDR Tx BWA3)
faults.49	ALARM	R/O	AlarmFlagRange R/O (IDR Tx BWA4)
internal.a	HEX	R/O	HexRange R/O (0 .. 0)
internal.b	HEX	R/O	HexRange R/O (0 .. 0)
internal.c	HEX	R/O	HexRange R/O (0 .. 0)
internal.d	HEX	R/O	HexRange R/O (0 .. 0)
internal.esc	TEXT	R/O	StringRange R/O
internal.rx.freq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
internal.tx.freq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.s	TEXT	R/O	StringRange R/O

9.139 Comtech-CDM760

This page describes the device driver and the device window for the Comtech CDM760 Modem via SNMP

It was developed with following SW version:

- Boot Loader version: 1.1.1
- App Loader version: 1.1.3
- Bulk version: 3.1.8

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The TX and modulation parameters.
- RX --- The Demodulator parameters.
- --- The carrier id settings.
- --- The acm and cnc settings.
- --- The interface settings.
- --- The gigabit statistics.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lnbPower	Defines if and which DC power the rx input port provides for lnb supply

parameter	description
lnbReference	Defines if reference is applied to the rx input port
lnbLowCurrent	Defines lower lnb current alarm limit
lnbHighCurrent	Defines upper lnb current alarm limit
bucReference	Defines if reference is applied to the tx output port
reference	Defines the used reference source
refAdjust	Adjust the reference source
piicSlot1	Defines the card type installed in piic slot 1
piicSlot2	Defines the card type installed in piic slot 2
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 13V 18V 24V)
config.lnbReference	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbLowCurrent	INTEGER	SETUP	IntegerRange (0 .. 500)
config.lnbHighCurrent	INTEGER	SETUP	IntegerRange (0 .. 500)
config.bucReference	CHOICE	SETUP	EnumRange (OFF ON)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXT-1MHZ EXT-2MHz EXT-5MHz EXT-10MHz INTERNAL+OUT)
config.refAdjust	INTEGER	SETUP	IntegerRange (-999 .. 999)
config.piicSlot1	CHOICE	SETUP	EnumRange (NONE ASI G703 OC3)

name	*type*	*flags*	*range*
config.piicSlot2	CHOICE	SETUP	EnumRange (NONE ASI G703 OC3)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.power	FLOAT		DoubleRange (-40.0 .. 0.0)
tx.mod.symbolRate	FLOAT		DoubleRange (0.100000 .. 150.000000)
tx.mod.datarate	FLOAT		DoubleRange (0.000000 .. 720.000000)
tx.mod.modcod	CHOICE		EnumRange (AUTO QPSK 1/2 QPSK 3/5 QPSK 2/3 QPSK 3/4 QPSK 4/5 QPSK 5/6 QPSK 8/9 QPSK 9/10 8PSK 3/5 8PSK 2/3 8PSK 3/4 8PSK 5/6 8PSK 8/9 8PSK 9/10 16APSK 2/3 16APSK 3/4 16APSK 4/5 16APSK 5/6 16APSK 8/9 16APSK 9/10 32APSK 3/4 32APSK 4/5 32APSK 5/6 32APSK 8/9 32APSK 9/10)
tx.mod.modulation	TEXT	R/O	StringRange R/O
tx.mod.fec	TEXT	R/O	StringRange R/O
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.rollOff	CHOICE		EnumRange (35% 25% 20% 15% 10% 5%)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)
tx.mod.frameSize	CHOICE		EnumRange (NORMAL SHORT)
tx.mod.goldCode	INTEGER		IntegerRange (0 .. 262141)
tx.meas.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.aupc.enable	CHOICE		EnumRange (OFF ON)
tx.aupc.esno	FLOAT		DoubleRange (0.0 .. 250.0)

name	*type*	*flags*	*range*
tx.aupc.nominal	FLOAT		DoubleRange (-40.0 .. 0.0)
tx.aupc.range	FLOAT		DoubleRange (0.0 .. 40.0)
tx.aupc.local	CHOICE		EnumRange (NOMINAL MAXIMUM)
tx.aupc.remote	CHOICE		EnumRange (NOMINAL MAXIMUM)
rx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
rx.mod.modcod	TEXT	R/O	StringRange R/O
rx.mod.standard	TEXT	R/O	StringRange R/O
rx.mod.modulation	TEXT	R/O	StringRange R/O
rx.mod.fec	TEXT	R/O	StringRange R/O
rx.mod.symbolRate	FLOAT		DoubleRange (0.100000 .. 150.000000)
rx.mod.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rx.mod.goldCode	INTEGER		IntegerRange (0 .. 262141)
rx.mod.esNoThreshold	FLOAT		DoubleRange (-3.0 .. 25.0)
rx.mod.spectrumInvert	TEXT	R/O	StringRange R/O
rx.mod.rollOff	TEXT	R/O	StringRange R/O
rx.mod.pilots	TEXT	R/O	StringRange R/O
rx.mod.frameSize	TEXT	R/O	StringRange R/O
rx.meas.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
mdm.payloadCompression	CHOICE		EnumRange (ON OFF)
mdm.payloadDecompression	CHOICE		EnumRange (ON OFF)
mdm.wanBufferLength	INTEGER		IntegerRange (20 .. 400)
mdm.hasPayloadGzipCard	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
mdm.standard	CHOICE		EnumRange (DVB-S2 DVB-S2-EB1 DVB-S2-EB2 DVB-S2X)
mdm.opMode	CHOICE		EnumRange (NORMAL MEO-TX-RX-HUB MEO-TX-RX-REMOTE MEO-RX-ONLY)
tx.mux.cid.enable	CHOICE	NOPRESET	EnumRange (OFF ON)
tx.mux.cid.longitude	TEXT	NOPRESET	StringRange
tx.mux.cid.latitude	TEXT	NOPRESET	StringRange
tx.mux.cid.telnNo	TEXT	NOPRESET	StringRange
tx.mux.cid.userInfo	TEXT	NOPRESET	StringRange
mdm.meoMode	TEXT	R/O	StringRange R/O
red.switchStandby	CHOICE		EnumRange (NORMAL FORCE)
red.switchState	TEXT	R/O	StringRange R/O
red.switchStatus	TEXT	R/O	StringRange R/O
red.control1toN	CHOICE		EnumRange (OFF ON)
red.protection	CHOICE		EnumRange (MANUAL AUTO)
cnc.enable	CHOICE		EnumRange (OFF ON)
cnc.minDelay	INTEGER		IntegerRange (0 .. 300)
cnc.maxDelay	INTEGER		IntegerRange (0 .. 300)
cnc.preDistortion	CHOICE		EnumRange (OFF ON)
acm.min.modcod	CHOICE		EnumRange (QPSK 1/2 QPSK 3/5 QPSK 2/3 QPSK 3/4 QPSK 4/5 QPSK 5/6 QPSK 8/9 QPSK 9/10 8PSK 3/5 8PSK 2/3 8PSK 3/4 8PSK 5/6 8PSK 8/9 8PSK 9/10 16APSK 2/3 16APSK 3/4 16APSK 4/5 16APSK 5/6 16APSK 8/9 16APSK 9/10 32APSK 3/4 32APSK 4/5 32APSK 5/6 32APSK 8/9 32APSK 9/10)

name	*type*	*flags*	*range*
acm.max.modcod	CHOICE		EnumRange (QPSK 1/2 QPSK 3/5 QPSK 2/3 QPSK 3/4 QPSK 4/5 QPSK 5/6 QPSK 8/9 QPSK 9/10 8PSK 3/5 8PSK 2/3 8PSK 3/4 8PSK 5/6 8PSK 8/9 8PSK 9/10 16APSK 2/3 16APSK 3/4 16APSK 4/5 16APSK 5/6 16APSK 8/9 16APSK 9/10 32APSK 3/4 32APSK 4/5 32APSK 5/6 32APSK 8/9 32APSK 9/10)
acm.demodUnlock	CHOICE		EnumRange (MINIMUM MODCOD MAINTAIN MODCOD)
acm.esnoCommon	INTEGER		IntegerRange (0 .. 55)
acm.esnoQpsk	INTEGER		IntegerRange (0 .. 99)
acm.esno8psk	INTEGER		IntegerRange (0 .. 99)
acm.esno16apsk	INTEGER		IntegerRange (0 .. 99)
acm.esno32apsk	INTEGER		IntegerRange (0 .. 99)
acm.esno64apsk	INTEGER		IntegerRange (0 .. 99)
acm.esno128apsk	INTEGER		IntegerRange (0 .. 99)
acm.esno256apsk	INTEGER		IntegerRange (0 .. 99)
acm.linearModCod	CHOICE		EnumRange (OFF ON)
mdm.circuitId	TEXT		StringRange
mdm.linkState.gigE1	TEXT	R/O	StringRange R/O
mdm.linkState.gigE2	TEXT	R/O	StringRange R/O
mdm.opticalEthernet	CHOICE		EnumRange (ON OFF)
mdm.udgEnable	CHOICE		EnumRange (ON OFF)
mdm.piic.1.tx.clock	CHOICE		EnumRange DIS (TX CLOCK EXTERNAL RX SAT)
mdm.piic.1.rx.clock	CHOICE		EnumRange DIS (TX CLOCK EXTERNAL RX SAT INTERNAL)

name	*type*	*flags*	*range*
mdm.piic.1.mode	CHOICE		EnumRange DIS (NONE E3 T3 STS1 STM)
mdm.piic.1.buffer	FLOAT		DoubleRange DIS (2.0 .. 76.0)
mdm.piic.2.tx.clock	CHOICE		EnumRange DIS (TX CLOCK EXTERNAL RX SAT)
mdm.piic.2.rx.clock	CHOICE		EnumRange DIS (TX CLOCK EXTERNAL RX SAT INTERNAL)
mdm.piic.2.mode	CHOICE		EnumRange DIS (NONE E3 T3 STS1 STM)
mdm.piic.2.buffer	FLOAT		DoubleRange DIS (2.0 .. 76.0)
tx.asi.1.mode	CHOICE		EnumRange DIS (OFF NORMAL ADVANCED)
tx.asi.1.framing	CHOICE		EnumRange DIS (NONE 188 204)
tx.asi.1.pcrRestamping	CHOICE		EnumRange DIS (OFF ON)
tx.asi.1.datarate	FLOAT		DoubleRange DIS (1.000000 .. 216.000000)
rx.asi.1.mode	CHOICE		EnumRange DIS (OFF NORMAL)
rx.asi.1.framing	CHOICE		EnumRange DIS (NONE 188 204)
rx.asi.1.outputType	CHOICE		EnumRange DIS (STREAMING BURST)
rx.asi.1.buffer	FLOAT		DoubleRange DIS (2.0 .. 76.0)
rx.asi.1.datarate	FLOAT		DoubleRange DIS (1.000000 .. 216.000000)
tx.asi.2.mode	CHOICE		EnumRange DIS (OFF NORMAL ADVANCED)
tx.asi.2.framing	CHOICE		EnumRange DIS (NONE 188 204)
tx.asi.2.pcrRestamping	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
tx.asi.2.datarate	FLOAT		DoubleRange DIS (1.000000 .. 216.000000)
rx.asi.2.mode	CHOICE		EnumRange DIS (OFF NORMAL)
rx.asi.2.framing	CHOICE		EnumRange DIS (NONE 188 204)
rx.asi.2.outputType	CHOICE		EnumRange DIS (STREAMING BURST)
rx.asi.2.buffer	FLOAT		DoubleRange DIS (2.0 .. 76.0)
rx.asi.2.datarate	FLOAT		DoubleRange DIS (1.000000 .. 216.000000)
tx.mod.cwMode	CHOICE		EnumRange (OFF CW SSB IQ-PN RF-LOOP IF-LOOP)
cnc.meas.ratio	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cnc.meas.delay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cnc.meas.freqOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
acm.tx.meas.modcod	TEXT	R/O	StringRange R/O
acm.rx.meas.modcod	TEXT	R/O	StringRange R/O
acm.rx.meas.remoteEsno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
acm.rx.meas.localEsno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.Lock	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.meas.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meas.modcod	TEXT	R/O	StringRange R/O
rx.meas.freqOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meas.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meas.wanBuffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meas.piic1Buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meas.piic2Buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.min.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
rx.meas.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meas.fer	TEXT	R/O	StringRange R/O
lnb.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
resetGigStats	TEXT	NOPRESET	StringRange
mdm.meas.gig.alias.01	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.02	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.03	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.04	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.05	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.06	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.07	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.08	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.09	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.10	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.11	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.12	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.13	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.14	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.15	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.16	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.17	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.18	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.19	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.20	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.21	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
mdm.meas.gig.alias.22	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.23	TEXT	R/O	StringRange R/O
mdm.meas.gig.alias.24	TEXT	R/O	StringRange R/O
mdm.meas.gig.count.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
mdm.meas.gig.count.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mdm.meas.gig.count.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.unit1	HEX	R/O	HexRange R/O (0 .. 0)
internal.unit2	HEX	R/O	HexRange R/O (0 .. 0)
internal.unit3	HEX	R/O	HexRange R/O (0 .. 0)
internal.unit4	HEX	R/O	HexRange R/O (0 .. 0)
internal.tx1	HEX	R/O	HexRange R/O (0 .. 0)
internal.tx2	HEX	R/O	HexRange R/O (0 .. 0)
internal.rx1	HEX	R/O	HexRange R/O (0 .. 0)
internal.rx2	HEX	R/O	HexRange R/O (0 .. 0)
internal.rx3	HEX	R/O	HexRange R/O (0 .. 0)
internal.oc3.id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.g703.id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 3.3V)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 1.5V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 1.2V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 1.8V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 1.0V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 2.5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Main Board PS 1.25V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 5.8V)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 3.8V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 2.8V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 3.3V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 1.5V)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Modem Board PS 1.2V)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power Calibration)
faults.15	ALARM	R/O	AlarmFlagRange R/O (FPGA Load)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Reference Lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit Temperature)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit Fault)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PIIC1 ExtClk Activity)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PIIC1 ExtClk Out Range)
faults.21	ALARM	R/O	AlarmFlagRange R/O (PIIC2 ExtClk Activity)
faults.22	ALARM	R/O	AlarmFlagRange R/O (PIIC2 ExtClk Out Range)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Log File)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Sys Clk PLL Lock)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Synth Clk 900MHz PLL)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (MEO Mdm to Mdm Comms)
faults.27	ALARM	R/O	AlarmFlagRange R/O (MEO Over Air Comms)
faults.28	ALARM	R/O	AlarmFlagRange R/O (MEO ACU Comms)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Ethernet Link)
faults.30	ALARM	R/O	AlarmFlagRange R/O (PaP Comm)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PaP Reset Req)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Sample Clk PLL Lock)
faults.33	ALARM	R/O	AlarmFlagRange R/O (LO Clk PLL Lock)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PIIC1 Sys Clk PLL)
faults.35	ALARM	R/O	AlarmFlagRange R/O (PIIC2 Sys Clk PLL)
faults.36	ALARM	R/O	AlarmFlagRange R/O (PIIC1 Input Signal Low)
faults.37	ALARM	R/O	AlarmFlagRange R/O (PIIC2 Input Signal Low)
faults.38	ALARM	R/O	AlarmFlagRange R/O (PIIC1 Frame Lock)
faults.39	ALARM	R/O	AlarmFlagRange R/O (PIIC2 Frame Lock)
faults.40	ALARM	R/O	AlarmFlagRange R/O (AUPC Power Level)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Demod Lock)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (Rx Level)
faults.43	ALARM	R/O	AlarmFlagRange R/O (CnC Lock)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Sample Clk PLL Lock)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Es/No Threshold)
faults.46	ALARM	R/O	AlarmFlagRange R/O (LO Clk PLL Lock)
faults.47	ALARM	R/O	AlarmFlagRange R/O (P1 Buffer Overflow)
faults.48	ALARM	R/O	AlarmFlagRange R/O (P1 Buffer Underflow)
faults.49	ALARM	R/O	AlarmFlagRange R/O (P1 Buffer Clock PLL)
faults.50	ALARM	R/O	AlarmFlagRange R/O (P2 Buffer Overflow)
faults.51	ALARM	R/O	AlarmFlagRange R/O (P2 Buffer Underflow)
faults.52	ALARM	R/O	AlarmFlagRange R/O (P2 Buffer Clock PLL)
faults.53	ALARM	R/O	AlarmFlagRange R/O (DeCompress Invalid)
faults.54	ALARM	R/O	AlarmFlagRange R/O (LNB Current)
faults.55	ALARM	R/O	AlarmFlagRange R/O (LNB Voltage)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Inv/Miss ACM Packet)
faults.57	ALARM	R/O	AlarmFlagRange R/O (DPD Lock)

9.140 Comtech-DT4500-SNMP

This page describes the device driver and the device window for the Comtech downconverter series DT4500 with Ethernet interface and support for the internal 1+N redundancy.

The driver detects the model name and adapts the frequency range.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.15vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.14vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vcc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.ifloTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.sumLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.coarseLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fineLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.converterNumber	TEXT	R/O	StringRange R/O
info.chainLength	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.autoFaultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTOMATIC)
config.coldStart	CHOICE	SETUP	EnumRange (ON OFF)
config.refFreqAdjustment	INTEGER	SETUP	IntegerRange (0 .. 0)
config.slopeAdjustment	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.redundancy	CHOICE	SETUP	EnumRange (ON OFF)
config.carrierMuteMode	CHOICE	SETUP	EnumRange (ON OFF)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.00 .. 25.00)
rx.attn	FLOAT		DoubleRange (0.00 .. 25.00)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
protection.01	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.02	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.03	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.04	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.05	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.06	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.07	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.08	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.09	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.10	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.11	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.12	CHOICE		EnumRange DIS (MANUAL AUTO)
state.01	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.02	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.03	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.04	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.05	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)

name	*type*	*flags*	*range*
state.06	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.07	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.08	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.09	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.10	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.11	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.12	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
freq.01	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.02	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.03	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.04	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.05	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.06	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.07	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.08	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.09	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
freq.10	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.11	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.12	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
attn.01	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.02	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.03	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.04	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.05	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.06	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.07	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.08	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.09	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.10	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.11	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.12	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
on.01	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.02	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.03	CHOICE	R/O	EnumRange R/O DIS (OFF ON)

name	*type*	*flags*	*range*
on.04	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.05	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.06	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.07	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.08	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.09	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.10	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.11	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.12	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
offs.01	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.02	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.03	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.04	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.05	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.06	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.07	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.08	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.09	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.10	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.11	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.12	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
location	CHOICE		EnumRange DIS (NONE 01)
redundancyMode	TEXT		StringRange DIS
reset	TEXT	NOPRESET	StringRange
internal.location	TEXT	R/O	StringRange R/O
faults.02	ALARM	R/O	AlarmFlagRange R/O (15 VDC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (14 VDC)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (VCC)
faults.05	ALARM	R/O	AlarmFlagRange R/O (IFLO Lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sum Lock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Coarse Lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fine Lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High Speed Bus)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Equalizer)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ext Reference)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switched)

9.141 Comtech-DT4500

This page describes the device driver and the device window for the Comtech downconverter series DT4500 with serial interface and support for the internal 1+1 redundancy.

The driver detects the model name and adapts the frequency range.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	*description*

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.15vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.14vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vcc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.ifloTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.sumLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.coarseLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fineLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.converterNumber	TEXT	R/O	StringRange R/O
info.chainLength	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.autoFaultRecovery	CHOICE	SETUP	EnumRange (ON OFF)
config.coldStart	CHOICE	SETUP	EnumRange (ON OFF)
config.refFreqAdjustment	INTEGER	SETUP	IntegerRange (0 .. 0)
config.slopeAdjustment	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.redundancy	CHOICE	SETUP	EnumRange (ON OFF)
config.freqLength	CHOICE	SAVE SETUP	EnumRange (4 5 7)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.00 .. 20.00)
rx.attn	FLOAT		DoubleRange (0.00 .. 20.00)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
state	CHOICE	R/O	EnumRange R/O (OFFLINE ONLINE)
location	CHOICE		EnumRange DIS (NONE)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
internal.oldloc	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (15 VDC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (14 VDC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VCC)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5VD)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (IFLO Lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sum Lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Coarse Lock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fine Lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High Speed Bus)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ref Lock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cold start)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Ext. Reference)

9.142 Comtech-LBC4000

This page describes the *Comtech-LBC4000* device driver and its device window. It controls 1 or 2 IF/L Upconverters or Downconverters via SNMP It supports 1:1 redundant as well as mixed configuration of one Up- and one Downconverter.

The number of converter modules and their type have to be defined in the device setup.

The driver was tested with one Downconverter and one Upconverter per Chassis.

- SW version 1.2.8

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Transmit and/or Receive control parameters.
- --- Redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
converterNumber	Select number of installed converters 1 or 2.
hasRedundancy	Select if unit is equipped with redundancy
converterOne	Select 1st converter type Upconverter or Downconverter.
converterTwo	Select 2nd converter type Upconverter or Downconverter.
lowPowerThdOne	Defines the low forward power threshold of converter 1.
lowPowerThdTwo	Defines the low forward power threshold of converter 2.
faultRecovery	Select Fault Recovery Mode Manual or Automatic
reference	Defines a reference frequency adjust value as Integer 0-255
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.psuA.12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuB.12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuA.8V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuB.8V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuA.5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuB.5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.refVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.reference	CHOICE	R/O	EnumRange R/O (N/A 5MHz 10MHz)

name	*type*	*flags*	*range*
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.converterNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
config.converterOne	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.converterTwo	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.hasRedundancy	CHOICE	SETUP	EnumRange (YES NO)
config.faultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.reference	INTEGER	SETUP	IntegerRange (0 .. 255)
config.deviceComment	TEXT	SAVE SETUP	StringRange
meas.temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.if.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.if.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.rf.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.rf.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
tx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
tx.on.1	CHOICE		EnumRange (OFF ON)
tx.on.2	CHOICE		EnumRange (OFF ON)
tx.frequency.1	FLOAT		DoubleRange (0.000 .. 0.000)
tx.frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
tx.slope.1	FLOAT		DoubleRange (-3.00 .. 3.00)
tx.slope.2	FLOAT		DoubleRange (-3.00 .. 3.00)

name	*type*	*flags*	*range*
rx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
rx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.on.2	CHOICE		EnumRange (OFF ON)
rx.frequency.1	FLOAT		DoubleRange (0.000 .. 0.000)
rx.frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
rx.slope.1	FLOAT		DoubleRange (-3.00 .. 3.00)
rx.slope.2	FLOAT		DoubleRange (-3.00 .. 3.00)
protection	CHOICE		EnumRange DIS (AUTO MANUAL)
position	CHOICE		EnumRange DIS (A B)
attnOffset	FLOAT		DoubleRange DIS (-3.00 .. 3.00)
clear	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU-A 12V)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU-A 8V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU-A 5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU-B 12V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU-B 8V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU-B 5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Conv.1 IF LO)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Conv.1 RF LO)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Conv.1 Temperature)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Conv.2 IF LO)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Conv.2 RF LO)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Conv.2 Temperature)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Ext. Ref)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX/TX Off)

9.143 Comtech-LPOD-SNMP

This page describes the device driver and the device window for the Comtech LPOD block upconverter over SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.faultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.lnbPower	CHOICE	SETUP	EnumRange (DISABLE NO-TONE 18V TONE-18V NO-TONE-13V TONE-13V)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.partno	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.amplifier	CHOICE		EnumRange (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.gain	FLOAT		DoubleRange (0.00 .. 20.00)
tx.attn	FLOAT		DoubleRange (0.00 .. 20.00)
reset	TEXT	NOPRESET	StringRange
protection	CHOICE		EnumRange (OFF 1+1 TX 1+1 TX/RX 1+2 TX 1+2 TX/RX MANUAL)
location	CHOICE	NOPRESET	EnumRange (ONLINE OFFLINE)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.on	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (BUC lock detect)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ref lock detect)
faults.03	ALARM	R/O	AlarmFlagRange R/O (FPGA done)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Flash checksum)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Forward power)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reverse power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Overtemp shutdown)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amplifier temp)
faults.09	ALARM	R/O	AlarmFlagRange R/O (FAN 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (FAN 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (24V power supply)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (24V switch power supply)
faults.13	ALARM	R/O	AlarmFlagRange R/O (13V power supply)
faults.14	ALARM	R/O	AlarmFlagRange R/O (7.8V power supply)
faults.15	ALARM	R/O	AlarmFlagRange R/O (5.8V power supply)
faults.16	ALARM	R/O	AlarmFlagRange R/O (2.5V power supply)
faults.17	ALARM	R/O	AlarmFlagRange R/O (1.2V power supply)
faults.18	ALARM	R/O	AlarmFlagRange R/O (-5.8V power supply)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX OFF)

9.144 Comtech-MBT5000

This page describes the *Comtech-MBT5000* device driver and its device window. It controls 2 Blockupconverters or 2 Blockdownconverters via SNMP as well as mixed configuration of one Up- and one Downconverter.

It was tested with a 2xBUC model

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Transmit and/or Receive control parameters.
- --- Redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- converterNumber --- Select number of installed converters 1 or 2.
- hasRedundancy --- Select if unit is equipped with redundancy
- converterOne --- Select 1st converter type UC odr DC
- converterTwo --- Select 2nd converter type UC odr DC
- faultRecovery --- Select Fault Recovery Mode Manual or Automatic
- reference --- Defines a reference frequency adjust value as Integer 0-255
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model.1	TEXT	R/O	StringRange R/O
info.model.2	TEXT	R/O	StringRange R/O
info.range.1	TEXT	R/O	StringRange R/O
info.range.2	TEXT	R/O	StringRange R/O
info.ps12V.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps12V.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps7.8V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps5V.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps5V.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.refVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.reference	CHOICE	R/O	EnumRange R/O (N/A 5MHz 10MHz)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.converterNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
config.hasRedundancy	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.converterOne	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.converterTwo	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.faultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.reference	INTEGER	SETUP	IntegerRange (0 .. 255)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
meas.temperature.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temperature.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faultCode.1	TEXT	R/O	StringRange R/O
faultCode.2	TEXT	R/O	StringRange R/O
meas.level.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.level.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
tx.gain.1	FLOAT		DoubleRange (0.00 .. 50.00)
tx.on.1	CHOICE		EnumRange (OFF ON)
tx.frequency.1	FLOAT		DoubleRange (0.000 .. 0.000)
tx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
tx.gain.2	FLOAT		DoubleRange (0.00 .. 50.00)
tx.on.2	CHOICE		EnumRange (OFF ON)
tx.frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
rx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
rx.gain.1	FLOAT		DoubleRange (0.00 .. 50.00)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
rx.gain.2	FLOAT		DoubleRange (0.00 .. 50.00)
rx.on.2	CHOICE		EnumRange (OFF ON)
protection	CHOICE		EnumRange DIS (AUTO MANUAL)
position	CHOICE		EnumRange DIS (A B)
tx.attnOffset.1	FLOAT		DoubleRange DIS (-3.00 .. 3.00)

name	*type*	*flags*	*range*
tx.attnOffset.2	FLOAT		DoubleRange DIS (-3.00 .. 3.00)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Converter 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Converter 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Base Unit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low Power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Insert)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Insert)
faults.08	ALARM	R/O	AlarmFlagRange R/O (LNA Current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX/TX Off)

9.145 Comtech-MBT5000A

This page describes the *Comtech-MBT5000A* device driver and its device window. It controls 2 Blockupconverters or 2 Blockdownconverters via SNMP as well as mixed configuration of one Up- and one Downconverter.

The number of converter modules and their type have to be defined in the device setup

It was tested with two Tri-Band Blockdownconverter in 1:1 redundant setup

- SW 1.1.4

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Transmit and/or Receive control parameters.
- --- Redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
converterNumber	Select number of installed converters 1 or 2.
converterOne	Select 1st converter type Upconverter or Downconverter.
converterTwo	Select 2nd converter type Upconverter or Downconverter.
hasRedundancy	Select if unit is equipped with redundancy
faultRecovery	Select Fault Recovery Mode Manual or Automatic
reference	Defines a reference frequency adjust value as Integer 0-255
controlPoint	Sets the actual remote control point (ETHERNET/FRONT PANEL/SERIAL)
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.modelConv.1	TEXT	R/O	StringRange R/O
info.modelConv.2	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.serialConv.1	TEXT	R/O	StringRange R/O
info.serialConv.2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.version	TEXT	R/O	StringRange R/O
info.range.1	TEXT	R/O	StringRange R/O
info.range.2	TEXT	R/O	StringRange R/O
info.ps19V.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps19V.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps8V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps5V.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.ps5V.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.refVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.reference	CHOICE	R/O	EnumRange R/O (N/A 5MHz 10MHz)
info.12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.faultCodeBase	TEXT	R/O	StringRange R/O
info.faultCodeUnit1	TEXT	R/O	StringRange R/O
info.faultCodeUnit2	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.converterNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
config.hasRedundancy	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.converterOne	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.converterTwo	CHOICE	SAVE SETUP	EnumRange (UPCONVERTER DOWNCONVERTER)
config.lowPowerThdOne	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.lowPowerThdTwo	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.faultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.reference	INTEGER	SETUP	IntegerRange (0 .. 255)
config.controlPoint	CHOICE	SETUP	EnumRange (ETHERNET FRONT PANEL SERIAL)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
int.unit1.bandString	TEXT	R/O	StringRange R/O
int.unit1.band1.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.band1.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.band2.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.band2.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.band3.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.band3.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit1.bandToFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.bandString	TEXT	R/O	StringRange R/O
int.unit2.band1.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.band1.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.band2.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.band2.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.band3.low	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.band3.high	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.unit2.bandToFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temperature.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temperature.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faultCode.1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faultCode.2	TEXT	R/O	StringRange R/O
meas.level.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.level.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
tx.gain.1	FLOAT		DoubleRange (0.00 .. 50.00)
tx.on.1	CHOICE		EnumRange (OFF ON)
tx.frequency.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
tx.band.1	CHOICE		EnumRange ()
tx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
tx.gain.2	FLOAT		DoubleRange (0.00 .. 50.00)
tx.on.2	CHOICE		EnumRange (OFF ON)
tx.frequency.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
tx.band.2	CHOICE		EnumRange ()
rx.attn.1	FLOAT		DoubleRange (0.00 .. 50.00)
rx.gain.1	FLOAT		DoubleRange (0.00 .. 50.00)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.frequency.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.band.1	CHOICE		EnumRange ()
rx.attn.2	FLOAT		DoubleRange (0.00 .. 50.00)
rx.gain.2	FLOAT		DoubleRange (0.00 .. 50.00)
rx.on.2	CHOICE		EnumRange (OFF ON)
rx.frequency.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.band.2	CHOICE		EnumRange ()
protection	CHOICE		EnumRange DIS (AUTO MANUAL)

name	*type*	*flags*	*range*
position	CHOICE		EnumRange DIS (A B)
tx.attnOffset.1	FLOAT		DoubleRange DIS (-3.00 .. 3.00)
tx.on	CHOICE		EnumRange DIS (OFF ON)
rx.on	CHOICE		EnumRange DIS (OFF ON)
rx.band	CHOICE		EnumRange DIS ()
tx.band	CHOICE		EnumRange DIS ()
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Converter 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Converter 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Base Unit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low Power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (External Reference)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Insert)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Insert)
faults.08	ALARM	R/O	AlarmFlagRange R/O (LNA Current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Remote Control)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX/TX Off)

9.146 Comtech-UT4500-SNMP

This page describes the device driver and the device window for the Comtech upconverter series UT4500 with Ethernet interface and support for the internal 1+N redundancy.

The driver detects the model name and adapts the frequency range.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.15vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.14vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vcc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.ifloTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.sumLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.coarseLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fineLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.converterNumber	TEXT	R/O	StringRange R/O
info.chainLength	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.autoFaultRecovery	CHOICE	SETUP	EnumRange (MANUAL AUTOMATIC)
config.coldStart	CHOICE	SETUP	EnumRange (ON OFF)
config.refFreqAdjustment	INTEGER	SETUP	IntegerRange (0 .. 0)
config.slopeAdjustment	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.redundancy	CHOICE	SETUP	EnumRange (ON OFF)
config.carrierMuteMode	CHOICE	SETUP	EnumRange (ON OFF)

name	*type*	*flags*	*range*
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.00 .. 25.00)
tx.attn	FLOAT		DoubleRange (0.00 .. 25.00)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
protection.01	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.02	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.03	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.04	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.05	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.06	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.07	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.08	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.09	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.10	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.11	CHOICE		EnumRange DIS (MANUAL AUTO)
protection.12	CHOICE		EnumRange DIS (MANUAL AUTO)
state.01	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.02	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.03	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.04	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.05	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.06	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)

name	*type*	*flags*	*range*
state.07	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.08	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.09	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.10	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.11	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
state.12	CHOICE	R/O	EnumRange R/O DIS (OFFLINE ONLINE)
freq.01	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.02	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.03	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.04	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.05	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.06	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.07	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.08	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.09	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.10	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
freq.11	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
freq.12	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
attn.01	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.02	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.03	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.04	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.05	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.06	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.07	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.08	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.09	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.10	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.11	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
attn.12	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 25.00)
on.01	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.02	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.03	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.04	CHOICE	R/O	EnumRange R/O DIS (OFF ON)

name	*type*	*flags*	*range*
on.05	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.06	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.07	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.08	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.09	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.10	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.11	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
on.12	CHOICE	R/O	EnumRange R/O DIS (OFF ON)
offs.01	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.02	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.03	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.04	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.05	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.06	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.07	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.08	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.09	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.10	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.11	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
offs.12	FLOAT		DoubleRange DIS (-20.00 .. 0.00)
location	CHOICE		EnumRange DIS (NONE 01)
redundancyMode	TEXT		StringRange DIS
reset	TEXT	NOPRESET	StringRange
internal.location	TEXT	R/O	StringRange R/O
faults.02	ALARM	R/O	AlarmFlagRange R/O (15 VDC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (14 VDC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VCC)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (IFLO Lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sum Lock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Coarse Lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fine Lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High Speed Bus)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Equalizer)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ext Reference)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switched)

9.147 Comtech-UT4500

This page describes the device driver and the device window for the Comtech upconverter series UT4500 with serial interface and support for the internal 1+1 redundancy.

The driver detects the model name and adapts the frequency range.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.15vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.14vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vcc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.ifloTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.sumLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.coarseLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fineLoopTuning	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.converterNumber	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.chainLength	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.autoFaultRecovery	CHOICE	SETUP	EnumRange (ON OFF)
config.coldStart	CHOICE	SETUP	EnumRange (ON OFF)
config.refFreqAdjustment	INTEGER	SETUP	IntegerRange (0 .. 0)
config.slopeAdjustment	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.redundancy	CHOICE	SETUP	EnumRange (ON OFF)
config.freqLength	CHOICE	SAVE SETUP	EnumRange (4 5 7)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.00 .. 25.00)
tx.attn	FLOAT		DoubleRange (0.00 .. 25.00)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
state	CHOICE	R/O	EnumRange R/O (OFFLINE ONLINE)
location	CHOICE		EnumRange DIS (NONE 01)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
internal.oldloc	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (15 VDC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (14 VDC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VCC)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (5VD)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (IFLO Lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sum Lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Coarse Lock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fine Lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High Speed Bus)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ref Lock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cold start)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Ext. Reference)

9.148 Conditional-Replicator

The *Conditional-Replicator* logical device replicates a parameter value from one device to a destination selected from a configured list of parameters. Selecting the destination is done by comparing a reference value read (typically from the source device) to reference values which are assigned to the destinations.

Copying the parameter value happens only on an operators request by clicking the 'REPLICATE' button. This button click also may be done programmatically by sending an arbitrary text to the 'take' variable of the *Conditional-Replicator*.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the 'REPLICATE' button.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the *Conditional-Replicator* device with the devices it shall connect.

parameter	***description***
sourceId	Enter the message identifier here of the parameter the *Conditional-Replicator* shall read and copy, e.g. "MODULATOR.tx.dataRate".
sourceRefId	Enter the message identifier here of the reference value the *Conditional-Replicator* shall read before copying to select the destination to copy to, e.g. "MATRIX.output.04".
destIds	Enter a comma separated list of message identifiers the *Conditional-Replicator* shall use as destinations. The list of reference values described below controls to which destinations the *Conditional-Replicator* actually copies the source value.
destRefValues	Enter a comma separated list of values here which define to which destination the device will copy its source value. If the reference value read from *sourceRefId* matches a value in this list, the source value is copied to the destination defined by the message identifier at the same position in the *destIds* list.

Remarks

- The *destIds* and *destRefValues* lists should not contain any space characters before or after the comma separators unless the space character is intended as a part of the (reference) value.
- The *destRefValues* list may contain the same value multiple times. In this case the *Conditional-Replicator* will copy its source value to multiple destinations if this value matches the reference value.
- In general TEXT or CHOICE type parameters should be used for the selection of the copy destinations. If you want to use a FLOAT for this, be sure to format the reference values in the *destRefValues* list in exactly the same way as the parameter would appear in the user interface. For comparison the *Conditional-Replicator* formats the floating point value read from *sourceRefId* following the range definition of this parameter and then compares this value (without unit string) to the values listed in *destRefValues*.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.sourceId	TEXT	SAVE SETUP	StringRange
config.sourceRefId	TEXT	SAVE SETUP	StringRange
config.destIds	TEXT	SAVE SETUP	StringRange
config.destRefValues	TEXT	SAVE SETUP	StringRange
take	TEXT	NOPRESET	StringRange

9.149 Cross-Frequency-Source

This page describes the device driver and the device window for the Cross Technology Frequency Source.

The driver was testet with model 2099-10 with options E (External Reference Input),

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter and some parameters to configure the unit for operation.

parameter	*description*
config.extRefFrequency	Select external reference type.
config.extRefOffset	Configure offset for external reference

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Cross-232 protocol. Communication is over a serial RS-232 interface possible.

Installation

Before you can operate the up-converter with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. You can configure the serial interface type of the unit at the front panel. (Select RS-232 via front panel)

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.extRefFrequency	CHOICE	SETUP	EnumRange DIS (1 MHz 5 MHz 10 MHz 25 MHz)
config.refOffset	INTEGER	SETUP	IntegerRange (-2000 .. 2000)
config.optionE	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
state.ext	BOOLEAN		BooleanRange DIS (true false)
state.int	BOOLEAN		BooleanRange (true false)
state.pll	BOOLEAN		BooleanRange DIS (true false)

name	*type*	*flags*	*range*
tx.level	INTEGER		IntegerRange (-10 .. 13)
tx.gain	INTEGER		IntegerRange DIS (-10 .. 10)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Oven Warm Up)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fault Occured)

9.150 Cross-UpDownConverter

This page describes the device driver and the device window for the Cross Technology Up/Down-converters.

The driver was testet with model 2017-02. The options X (125kHz Freq. Step), L (LNB current) and V (SSPB current) are not supported by this driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter and some parameters to configure the unit for operation.

parameter	*description*
10MHzRef	Enable external 10MHz reference
config.tx.insertRef	Enable 10MHz reference for upconverter
config.rx.insertRef	Enable 10MHz reference for downconverter

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Cross-232 protocol. Communication is over a serial RS-232 interface possible.

Installation

Before you can operate the up-converter with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. You can configure the serial interface of the unit at the front panel.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.tx.insertRef	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.insertRef	CHOICE	SETUP	EnumRange (OFF ON)
config.10MHzRef	CHOICE	SETUP	EnumRange (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	INTEGER		IntegerRange (-10 .. 30)
tx.frequency	INTEGER		IntegerRange (950 .. 2150)

name	*type*	*flags*	*range*
rx.gain	INTEGER		IntegerRange (0 .. 50)
rx.frequency	INTEGER		IntegerRange (950 .. 2150)
rx.invert	CHOICE		EnumRange (NORM INV)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Rx synthesizer)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Tx synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.151 DEV-4111-DA

This page describes the device driver and the device window for DEV 411 Intelligent Universal Optribution Chassis; 2 Slots with 1:1 redundant amplifier configuration via SNMP

Tested with:

- Firmware: 32-0078 O.01
- Switch Module: DEV 11-0088
- Amp Module: DEV 11-0096

The driver detects the feature set of up to 2 cards installed in the 4111 and en/disables it´s variables accordingly.

The driver reads out the number of installed modules and sets its parameters accordingly.
Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The ports control and statistics parameters.
- --- The switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
info.noSwGroups	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noSwUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.noPorts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noLines	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noPsu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noFans	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
ctrlMode	CHOICE		EnumRange (LOCAL REMOTE)
position	CHOICE		EnumRange DIS (A B)
port.01.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.rfLimit	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.rfLimit	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.enable	CHOICE		EnumRange DIS (ON OFF)
port.02.enable	CHOICE		EnumRange DIS (ON OFF)
port.01.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
port.02.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
port.01.biasLowLimit	FLOAT		DoubleRange DIS (0.0 .. 300.0)
port.02.biasLowLimit	FLOAT		DoubleRange DIS (0.0 .. 300.0)
port.01.biasHighLimit	FLOAT		DoubleRange DIS (250.0 .. 600.0)
port.02.biasHighLimit	FLOAT		DoubleRange DIS (250.0 .. 600.0)
port.01.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.gainMode	CHOICE		EnumRange DIS (MANUAL AUTO MUTE)

name	*type*	*flags*	*range*
port.02.gainMode	CHOICE		EnumRange DIS (MANUAL AUTO MUTE)
port.01.tilt	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.tilt	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.inputPwr	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.inputPwr	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.inputPwrLimit	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.inputPwrLimit	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.outputPwr	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.outputPwr	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.outputPwrSetpoint	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.outputPwrSetpoint	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.name	TEXT	SAVE	StringRange
port.02.name	TEXT	SAVE	StringRange
internal.hasBias	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasGain	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasTilt	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasRfSense	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasOptics	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasInputPwr	CHOICE	R/O	EnumRange R/O (YES NO)
internal.hasOutputPwr	CHOICE	R/O	EnumRange R/O (YES NO)
internal.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Line 1)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Line 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Bus & Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Monitoring Card)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Port 1 State)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Port 1 RF Level)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Port 1 Bias)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Port 1 Optics)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Port 2 State)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Port 2 RF Level)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Port 2 Bias)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Port 2 Optics)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Remote access)

9.152 DEV-Distribution-2190

This page describes the device driver and the device window for DEV 2190 managed L-Band distribution system with up to 16 amplifier modules type DEV 11-0096. It includes level and alarm monitoring for rf and bias current without adjustment of the limits. Each port can be assigned with an sat-nms internal port name.

The driver does not support mixed configurations and redundancy switching.

The driver was tested with

- Firmware: 32-0059 Q.03
- Module type: DEV 11-0096

The driver reads out the number of installed modules and sets its parameters accordingly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main frame parameters page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- port.xx.biasEnable --- Defines if the respective ports should supply a bias voltage

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.numberOfPorts	INTEGER	R/O	IntegerRange R/O (0 .. 16)
info.numberOfPs	INTEGER	R/O	IntegerRange R/O (0 .. 3)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.port.01.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.02.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.03.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.04.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.05.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.06.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.07.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.08.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.09.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.10.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.11.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.12.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.13.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.14.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.port.15.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)

name	*type*	*flags*	*range*
config.port.16.biasEnable	CHOICE	SETUP	EnumRange DIS (ON OFF)
ctrlMode	CHOICE		EnumRange (LOCAL REMOTE)
port.01.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.03.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.04.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.05.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.06.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.07.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.08.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.09.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.10.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.11.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.12.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.13.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.14.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.15.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.16.level	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.enable	CHOICE		EnumRange DIS (ON OFF)
port.02.enable	CHOICE		EnumRange DIS (ON OFF)
port.03.enable	CHOICE		EnumRange DIS (ON OFF)
port.04.enable	CHOICE		EnumRange DIS (ON OFF)
port.05.enable	CHOICE		EnumRange DIS (ON OFF)
port.06.enable	CHOICE		EnumRange DIS (ON OFF)
port.07.enable	CHOICE		EnumRange DIS (ON OFF)
port.08.enable	CHOICE		EnumRange DIS (ON OFF)

name	*type*	*flags*	*range*
port.09.enable	CHOICE		EnumRange DIS (ON OFF)
port.10.enable	CHOICE		EnumRange DIS (ON OFF)
port.11.enable	CHOICE		EnumRange DIS (ON OFF)
port.12.enable	CHOICE		EnumRange DIS (ON OFF)
port.13.enable	CHOICE		EnumRange DIS (ON OFF)
port.14.enable	CHOICE		EnumRange DIS (ON OFF)
port.15.enable	CHOICE		EnumRange DIS (ON OFF)
port.16.enable	CHOICE		EnumRange DIS (ON OFF)
port.01.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.02.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.03.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.04.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.05.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.06.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.07.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.08.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.09.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.10.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.11.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.12.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.13.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.14.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.15.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.16.bias	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
port.01.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.02.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
port.03.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.04.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.05.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.06.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.07.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.08.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.09.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.10.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.11.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.12.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.13.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.14.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.15.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.16.gain	FLOAT		DoubleRange DIS (0.0 .. 0.0)
port.01.name	TEXT	SAVE	StringRange
port.02.name	TEXT	SAVE	StringRange
port.03.name	TEXT	SAVE	StringRange
port.04.name	TEXT	SAVE	StringRange
port.05.name	TEXT	SAVE	StringRange
port.06.name	TEXT	SAVE	StringRange
port.07.name	TEXT	SAVE	StringRange
port.08.name	TEXT	SAVE	StringRange
port.09.name	TEXT	SAVE	StringRange
port.10.name	TEXT	SAVE	StringRange
port.11.name	TEXT	SAVE	StringRange
port.12.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
port.13.name	TEXT	SAVE	StringRange
port.14.name	TEXT	SAVE	StringRange
port.15.name	TEXT	SAVE	StringRange
port.16.name	TEXT	SAVE	StringRange
internal.01.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.02.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.03.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.04.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.05.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.06.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.07.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.08.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.09.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.10.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.11.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.12.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.13.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.14.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.15.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.16.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Line 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Line 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Bus & Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Monitoring Card)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Port 1 State)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Port 1 RF Level)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Port 1 Bias)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Port 2 State)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Port 2 RF Level)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Port 2 Bias)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Port 3 State)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Port 3 RF Level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Port 3 Bias)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Port 4 State)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Port 4 RF Level)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Port 4 Bias)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Port 5 State)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Port 5 RF Level)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Port 5 Bias)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Port 6 State)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Port 6 RF Level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Port 6 Bias)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Port 7 State)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (Port 7 RF Level)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Port 7 Bias)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Port 8 State)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Port 8 RF Level)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Port 8 Bias)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Port 9 State)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Port 9 RF Level)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Port 9 Bias)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Port 10 State)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Port 10 RF Level)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Port 10 Bias)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Port 11 State)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Port 11 RF Level)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Port 11 Bias)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Port 12 State)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Port 12 RF Level)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Port 12 Bias)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Port 13 State)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Port 13 RF Level)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Port 13 Bias)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Port 14 State)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Port 14 RF Level)

name	*type*	*flags*	*range*
faults.50	ALARM	R/O	AlarmFlagRange R/O (Port 14 Bias)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Port 15 State)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Port 15 RF Level)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Port 15 Bias)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Port 16 State)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Port 16 RF Level)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Port 16 Bias)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Remote access)

9.153 DEV-FO-RX

This page describes the **DEV-FO-RX** device driver and its device window. The**DEV-FO-RX** device driver monitors and controls one single or dual channel FO receive module card.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the DEV chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:

- RX module 7332 within DEV 7114 chassis

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info c.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
- slot --- Selects the physical slot where the module is fitted in the chassis to control.
- useId --- Defines is the port indexes are calculated automatically based on the number of available modules. Using AUTO should work on all frames using a common card type in all slots. If AUTO selection is not matching use the DEV Frame driver overview to determine port indexes.
- idChannelA --- Sets the port index of channel A in Manual Mode.
- idChannelB --- Sets the port index of channel B in Manual Mode.
- dualChannel --- Defines if the unit is single or dual channel module.
- monitoring --- En/Disable power monitoring if applicable.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.noPorts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.module	TEXT	R/O	StringRange R/O
info.hasRfSense	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasGain	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasTilt	CHOICE	R/O	EnumRange R/O (YES NO)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
config.useId	CHOICE	SAVE SETUP	EnumRange (AUTO MANUAL)
config.idChannelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.idChannelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.dualChannel	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.monitoring	CHOICE	SETUP	EnumRange (ENABLED DISABLED)

name	*type*	*flags*	*range*
ctrlMode	CHOICE		EnumRange (LOCAL REMOTE)
channel.a.rfPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.b.rfPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.a.rfLimit	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.rfLimit	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.monitoring	CHOICE		EnumRange (ENABLED DISABLED)
channel.b.monitoring	CHOICE		EnumRange (ENABLED DISABLED)
channel.a.gainMode	CHOICE		EnumRange (MANUAL AUTO MUTE)
channel.b.gainMode	CHOICE		EnumRange (MANUAL AUTO MUTE)
channel.a.gain	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.gain	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.setpoint	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.setpoint	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.tilt	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.tilt	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Card)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Ch. A RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ch. B RF)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Ch. A Fiber)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Ch. B Fiber)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Surveillance)

9.154 DEV-FO-TX

This page describes the **DEV-FO-TX** device driver and its device window. The**DEV-FO-TX** device driver monitors and controls one single or dual channel FO transmit module card.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of

the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the DEV chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:

- Twin TX 7241 module within DEV 1752 outdoor chassis
- TX module DEV 7233 within DEV 7713 chassis

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info c.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip

which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

- slot --- Selects the physical slot where the module is fitted in the chassis to control.
- useId --- Defines is the port indexes are calculated automatically based on the number of available modules. Using AUTO should work on all frames using a common card type in all slots. If AUTO selection is not matching use the DEV Frame driver overview to determine port indexes.
- idChannelA --- Sets the port index of channel A in Manual Mode.
- idChannelB --- Sets the port index of channel B in Manual Mode.
- dualChannel --- Defines if the unit is single or dual channel module.
- monitoring --- En/Disable power monitoring if applicable.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.noPorts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.module	TEXT	R/O	StringRange R/O
info.hasRfSense	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasGain	CHOICE	R/O	EnumRange R/O (YES NO)

name	*type*	*flags*	*range*
info.hasTilt	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasBias	CHOICE	R/O	EnumRange R/O (YES NO)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
config.useId	CHOICE	SAVE SETUP	EnumRange (AUTO MANUAL)
config.idChannelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.idChannelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.dualChannel	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.monitoring	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
ctrlMode	CHOICE		EnumRange (LOCAL REMOTE)
channel.a.rfPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.b.rfPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.a.rfLimit	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.rfLimit	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.monitoring	CHOICE		EnumRange (ENABLED DISABLED)
channel.b.monitoring	CHOICE		EnumRange (ENABLED DISABLED)
channel.a.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.b.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
channel.a.biasEnable	CHOICE		EnumRange (OFF VL HL VH HH)
channel.b.biasEnable	CHOICE		EnumRange (OFF VL HL VH HH)
channel.a.biasLowLimit	FLOAT		DoubleRange (0.0 .. 300.0)
channel.b.biasLowLimit	FLOAT		DoubleRange (0.0 .. 300.0)
channel.a.biasHighLimit	FLOAT		DoubleRange (250.0 .. 600.0)

name	*type*	*flags*	*range*
channel.b.biasHighLimit	FLOAT		DoubleRange (250.0 .. 600.0)
channel.a.gainMode	CHOICE		EnumRange (MANUAL AUTO MUTE)
channel.b.gainMode	CHOICE		EnumRange (MANUAL AUTO MUTE)
channel.a.gain	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.gain	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.setpoint	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.setpoint	FLOAT		DoubleRange (0.0 .. 0.0)
channel.a.tilt	FLOAT		DoubleRange (0.0 .. 0.0)
channel.b.tilt	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Card)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Ch. A RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ch. B RF)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ch. A Bias)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Ch. B Bias)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Ch. A Fiber)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Ch. B Fiber)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Surveillance)

9.155 DEV-Frame

This page describes the universal *DEV-Frame* device driver and its device window. The *DEV-Frame* device driver monitors and controls the base frame/unit features and alarms of one DEV chassis. The frame driver supports frames with up to 20 modules using the dev-switch MIB.

Beside this base unit multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The driver was tested with the following equipment:

- DEV 1953 Universal Switch Distribution Chassis 3RU (up to 16 modules)
- DEV 7152 Intelligent Outdoor Optribution Chassis
- DEV 7114 Intelligent Optribution Chassis 4 RU (up to 16 modules)
- DEV 7113 Intelligent Optribution Chassis; 20 Slots
- SW version: e567 1v9

The different available optional cards of the ETL Genus series are:

- DEV-FO-TX monitors and controls one single or dual channel FO transmit module
- DEV-FO-RX monitors and controls one single or dual channel FO receive module
- DEV-Switch-2-To-1 monitors and controls one 1:2 Optical Switch Module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules port types and their sources.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.noSwGroups	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noSwUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noPorts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noLines	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noPsu	INTEGER	R/O	IntegerRange R/O (0 .. 3)
info.noFans	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.hasAutoMode	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.deviceComment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ctrlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE AUTO)
config.swMode	CHOICE	SETUP	EnumRange (SIMULTANEOUS INDIVIDUAL)
port.01.type	TEXT	R/O	StringRange R/O DIS
port.02.type	TEXT	R/O	StringRange R/O DIS
port.03.type	TEXT	R/O	StringRange R/O DIS
port.04.type	TEXT	R/O	StringRange R/O DIS
port.05.type	TEXT	R/O	StringRange R/O DIS
port.06.type	TEXT	R/O	StringRange R/O DIS
port.07.type	TEXT	R/O	StringRange R/O DIS
port.08.type	TEXT	R/O	StringRange R/O DIS
port.09.type	TEXT	R/O	StringRange R/O DIS
port.10.type	TEXT	R/O	StringRange R/O DIS
port.11.type	TEXT	R/O	StringRange R/O DIS
port.12.type	TEXT	R/O	StringRange R/O DIS
port.13.type	TEXT	R/O	StringRange R/O DIS
port.14.type	TEXT	R/O	StringRange R/O DIS
port.15.type	TEXT	R/O	StringRange R/O DIS
port.16.type	TEXT	R/O	StringRange R/O DIS
port.17.type	TEXT	R/O	StringRange R/O DIS
port.18.type	TEXT	R/O	StringRange R/O DIS
port.19.type	TEXT	R/O	StringRange R/O DIS
port.20.type	TEXT	R/O	StringRange R/O DIS
port.21.type	TEXT	R/O	StringRange R/O DIS
port.22.type	TEXT	R/O	StringRange R/O DIS
port.23.type	TEXT	R/O	StringRange R/O DIS
port.24.type	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
port.25.type	TEXT	R/O	StringRange R/O DIS
port.26.type	TEXT	R/O	StringRange R/O DIS
port.27.type	TEXT	R/O	StringRange R/O DIS
port.28.type	TEXT	R/O	StringRange R/O DIS
port.29.type	TEXT	R/O	StringRange R/O DIS
port.30.type	TEXT	R/O	StringRange R/O DIS
port.31.type	TEXT	R/O	StringRange R/O DIS
port.32.type	TEXT	R/O	StringRange R/O DIS
port.33.type	TEXT	R/O	StringRange R/O DIS
port.34.type	TEXT	R/O	StringRange R/O DIS
port.35.type	TEXT	R/O	StringRange R/O DIS
port.36.type	TEXT	R/O	StringRange R/O DIS
port.37.type	TEXT	R/O	StringRange R/O DIS
port.38.type	TEXT	R/O	StringRange R/O DIS
port.39.type	TEXT	R/O	StringRange R/O DIS
port.40.type	TEXT	R/O	StringRange R/O DIS
port.01.src	TEXT	R/O	StringRange R/O DIS
port.02.src	TEXT	R/O	StringRange R/O DIS
port.03.src	TEXT	R/O	StringRange R/O DIS
port.04.src	TEXT	R/O	StringRange R/O DIS
port.05.src	TEXT	R/O	StringRange R/O DIS
port.06.src	TEXT	R/O	StringRange R/O DIS
port.07.src	TEXT	R/O	StringRange R/O DIS
port.08.src	TEXT	R/O	StringRange R/O DIS
port.09.src	TEXT	R/O	StringRange R/O DIS
port.10.src	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
port.11.src	TEXT	R/O	StringRange R/O DIS
port.12.src	TEXT	R/O	StringRange R/O DIS
port.13.src	TEXT	R/O	StringRange R/O DIS
port.14.src	TEXT	R/O	StringRange R/O DIS
port.15.src	TEXT	R/O	StringRange R/O DIS
port.16.src	TEXT	R/O	StringRange R/O DIS
port.17.src	TEXT	R/O	StringRange R/O DIS
port.18.src	TEXT	R/O	StringRange R/O DIS
port.19.src	TEXT	R/O	StringRange R/O DIS
port.20.src	TEXT	R/O	StringRange R/O DIS
port.21.src	TEXT	R/O	StringRange R/O DIS
port.22.src	TEXT	R/O	StringRange R/O DIS
port.23.src	TEXT	R/O	StringRange R/O DIS
port.24.src	TEXT	R/O	StringRange R/O DIS
port.25.src	TEXT	R/O	StringRange R/O DIS
port.26.src	TEXT	R/O	StringRange R/O DIS
port.27.src	TEXT	R/O	StringRange R/O DIS
port.28.src	TEXT	R/O	StringRange R/O DIS
port.29.src	TEXT	R/O	StringRange R/O DIS
port.30.src	TEXT	R/O	StringRange R/O DIS
port.31.src	TEXT	R/O	StringRange R/O DIS
port.32.src	TEXT	R/O	StringRange R/O DIS
port.33.src	TEXT	R/O	StringRange R/O DIS
port.34.src	TEXT	R/O	StringRange R/O DIS
port.35.src	TEXT	R/O	StringRange R/O DIS
port.36.src	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
port.37.src	TEXT	R/O	StringRange R/O DIS
port.38.src	TEXT	R/O	StringRange R/O DIS
port.39.src	TEXT	R/O	StringRange R/O DIS
port.40.src	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power line 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power line 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Support hardware)

9.156 DEV-Matrix-1986

This page describes the device driver and the device window for a DEV This page describes the device driver and the device window for a DEV 1986 L-Band Matrix with up to 64 inputs and outputs. If the number of interfaces changes it requires restarting of the M&C service, because the the interface number is only read one time at startup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port Switching by Alias Name 1-32.
- abc --- Port Switching by Alias Name 33-64.
- 123 --- Port Switching by Number 1-32.
- 123 --- Port Switching by Number 33-64.
- --- Edit the Input Alias Names.
- --- Edit the Output Alias Names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DEV 1986 matrix.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Delays the read back of a crosspoint setting actually after it has been set by the driver to at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.ctrlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
info.deviceType	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.mibRevision	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.powerSupplies	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.hasRfSense	CHOICE	R/O	EnumRange R/O (YES NO)
output.01	CHOICE		EnumRange (00)
output.02	CHOICE		EnumRange DIS (00)
output.03	CHOICE		EnumRange DIS (00)
output.04	CHOICE		EnumRange DIS (00)
output.05	CHOICE		EnumRange DIS (00)
output.06	CHOICE		EnumRange DIS (00)

name	*type*	*flags*	*range*
output.07	CHOICE		EnumRange DIS (00)
output.08	CHOICE		EnumRange DIS (00)
output.09	CHOICE		EnumRange DIS (00)
output.10	CHOICE		EnumRange DIS (00)
output.11	CHOICE		EnumRange DIS (00)
output.12	CHOICE		EnumRange DIS (00)
output.13	CHOICE		EnumRange DIS (00)
output.14	CHOICE		EnumRange DIS (00)
output.15	CHOICE		EnumRange DIS (00)
output.16	CHOICE		EnumRange DIS (00)
output.17	CHOICE		EnumRange DIS (00)
output.18	CHOICE		EnumRange DIS (00)
output.19	CHOICE		EnumRange DIS (00)
output.20	CHOICE		EnumRange DIS (00)
output.21	CHOICE		EnumRange DIS (00)
output.22	CHOICE		EnumRange DIS (00)
output.23	CHOICE		EnumRange DIS (00)
output.24	CHOICE		EnumRange DIS (00)
output.25	CHOICE		EnumRange DIS (00)
output.26	CHOICE		EnumRange DIS (00)
output.27	CHOICE		EnumRange DIS (00)
output.28	CHOICE		EnumRange DIS (00)
output.29	CHOICE		EnumRange DIS (00)
output.30	CHOICE		EnumRange DIS (00)
output.31	CHOICE		EnumRange DIS (00)
output.32	CHOICE		EnumRange DIS (00)
output.33	CHOICE		EnumRange DIS (00)

name	*type*	*flags*	*range*
output.34	CHOICE		EnumRange DIS (00)
output.35	CHOICE		EnumRange DIS (00)
output.36	CHOICE		EnumRange DIS (00)
output.37	CHOICE		EnumRange DIS (00)
output.38	CHOICE		EnumRange DIS (00)
output.39	CHOICE		EnumRange DIS (00)
output.40	CHOICE		EnumRange DIS (00)
output.41	CHOICE		EnumRange DIS (00)
output.42	CHOICE		EnumRange DIS (00)
output.43	CHOICE		EnumRange DIS (00)
output.44	CHOICE		EnumRange DIS (00)
output.45	CHOICE		EnumRange DIS (00)
output.46	CHOICE		EnumRange DIS (00)
output.47	CHOICE		EnumRange DIS (00)
output.48	CHOICE		EnumRange DIS (00)
output.49	CHOICE		EnumRange DIS (00)
output.50	CHOICE		EnumRange DIS (00)
output.51	CHOICE		EnumRange DIS (00)
output.52	CHOICE		EnumRange DIS (00)
output.53	CHOICE		EnumRange DIS (00)
output.54	CHOICE		EnumRange DIS (00)
output.55	CHOICE		EnumRange DIS (00)
output.56	CHOICE		EnumRange DIS (00)
output.57	CHOICE		EnumRange DIS (00)
output.58	CHOICE		EnumRange DIS (00)

name	*type*	*flags*	*range*
output.59	CHOICE		EnumRange DIS (00)
output.60	CHOICE		EnumRange DIS (00)
output.61	CHOICE		EnumRange DIS (00)
output.62	CHOICE		EnumRange DIS (00)
output.63	CHOICE		EnumRange DIS (00)
output.64	CHOICE		EnumRange DIS (00)
output.65	CHOICE		EnumRange DIS (00)
byname.01	CHOICE		EnumRange (00)
byname.02	CHOICE		EnumRange DIS (00)
byname.03	CHOICE		EnumRange DIS (00)
byname.04	CHOICE		EnumRange DIS (00)
byname.05	CHOICE		EnumRange DIS (00)
byname.06	CHOICE		EnumRange DIS (00)
byname.07	CHOICE		EnumRange DIS (00)
byname.08	CHOICE		EnumRange DIS (00)
byname.09	CHOICE		EnumRange DIS (00)
byname.10	CHOICE		EnumRange DIS (00)
byname.11	CHOICE		EnumRange DIS (00)
byname.12	CHOICE		EnumRange DIS (00)
byname.13	CHOICE		EnumRange DIS (00)
byname.14	CHOICE		EnumRange DIS (00)
byname.15	CHOICE		EnumRange DIS (00)
byname.16	CHOICE		EnumRange DIS (00)
byname.17	CHOICE		EnumRange DIS (00)
byname.18	CHOICE		EnumRange DIS (00)
byname.19	CHOICE		EnumRange DIS (00)
byname.20	CHOICE		EnumRange DIS (00)

name	*type*	*flags*	*range*
byname.21	CHOICE		EnumRange DIS (00)
byname.22	CHOICE		EnumRange DIS (00)
byname.23	CHOICE		EnumRange DIS (00)
byname.24	CHOICE		EnumRange DIS (00)
byname.25	CHOICE		EnumRange DIS (00)
byname.26	CHOICE		EnumRange DIS (00)
byname.27	CHOICE		EnumRange DIS (00)
byname.28	CHOICE		EnumRange DIS (00)
byname.29	CHOICE		EnumRange DIS (00)
byname.30	CHOICE		EnumRange DIS (00)
byname.31	CHOICE		EnumRange DIS (00)
byname.32	CHOICE		EnumRange DIS (00)
byname.33	CHOICE		EnumRange DIS (00)
byname.34	CHOICE		EnumRange DIS (00)
byname.35	CHOICE		EnumRange DIS (00)
byname.36	CHOICE		EnumRange DIS (00)
byname.37	CHOICE		EnumRange DIS (00)
byname.38	CHOICE		EnumRange DIS (00)
byname.39	CHOICE		EnumRange DIS (00)
byname.40	CHOICE		EnumRange DIS (00)
byname.41	CHOICE		EnumRange DIS (00)
byname.42	CHOICE		EnumRange DIS (00)
byname.43	CHOICE		EnumRange DIS (00)
byname.44	CHOICE		EnumRange DIS (00)
byname.45	CHOICE		EnumRange DIS (00)

name	*type*	*flags*	*range*
byname.46	CHOICE		EnumRange DIS (00)
byname.47	CHOICE		EnumRange DIS (00)
byname.48	CHOICE		EnumRange DIS (00)
byname.49	CHOICE		EnumRange DIS (00)
byname.50	CHOICE		EnumRange DIS (00)
byname.51	CHOICE		EnumRange DIS (00)
byname.52	CHOICE		EnumRange DIS (00)
byname.53	CHOICE		EnumRange DIS (00)
byname.54	CHOICE		EnumRange DIS (00)
byname.55	CHOICE		EnumRange DIS (00)
byname.56	CHOICE		EnumRange DIS (00)
byname.57	CHOICE		EnumRange DIS (00)
byname.58	CHOICE		EnumRange DIS (00)
byname.59	CHOICE		EnumRange DIS (00)
byname.60	CHOICE		EnumRange DIS (00)
byname.61	CHOICE		EnumRange DIS (00)
byname.62	CHOICE		EnumRange DIS (00)
byname.63	CHOICE		EnumRange DIS (00)
byname.64	CHOICE		EnumRange DIS (00)
byname.65	CHOICE		EnumRange DIS (00)
output.level.01	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.02	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.03	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.04	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.05	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.06	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.07	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
output.level.08	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.09	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.10	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.11	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.12	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.13	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.14	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.15	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.16	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.17	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.18	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.19	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.20	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.21	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.22	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.23	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.24	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.25	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.26	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.27	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.28	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.29	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.30	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.31	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.32	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
output.level.33	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.34	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.35	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.36	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.37	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.38	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.39	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.40	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.41	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.42	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.43	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.44	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.45	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.46	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.47	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.48	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.49	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.50	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.51	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.52	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.53	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.54	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.55	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.56	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.57	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.58	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
output.level.59	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.60	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.61	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.62	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.63	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.level.64	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.state.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.11	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.12	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.13	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.14	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.15	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.16	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.17	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.18	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.19	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.20	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.21	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
output.state.22	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.23	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.24	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.25	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.26	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.27	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.28	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.29	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.30	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.31	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.32	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.33	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.34	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.35	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.36	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.37	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.38	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.39	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.40	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.41	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.42	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.43	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.44	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.45	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.46	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
output.state.47	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.48	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.49	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.50	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.51	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.52	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.53	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.54	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.55	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.56	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.57	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.58	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.59	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.60	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.61	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.62	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.63	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.state.64	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
output.card.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.11	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.12	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.13	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.14	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.15	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.16	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.17	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.18	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.19	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.20	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.21	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.22	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.23	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.24	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.25	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.26	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.27	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.28	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.29	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.30	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.31	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.32	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.33	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.34	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
output.card.35	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.36	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.37	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.38	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.39	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.40	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.41	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.42	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.43	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.44	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.45	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.46	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.47	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.48	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.49	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.50	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.51	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.52	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.53	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.54	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.55	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.56	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.57	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.58	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.59	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.60	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
output.card.61	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.62	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.63	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.card.64	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.name.01	TEXT		StringRange
input.name.02	TEXT		StringRange
input.name.03	TEXT		StringRange
input.name.04	TEXT		StringRange
input.name.05	TEXT		StringRange
input.name.06	TEXT		StringRange
input.name.07	TEXT		StringRange
input.name.08	TEXT		StringRange
input.name.09	TEXT		StringRange
input.name.10	TEXT		StringRange
input.name.11	TEXT		StringRange
input.name.12	TEXT		StringRange
input.name.13	TEXT		StringRange
input.name.14	TEXT		StringRange
input.name.15	TEXT		StringRange
input.name.16	TEXT		StringRange
input.name.17	TEXT		StringRange
input.name.18	TEXT		StringRange
input.name.19	TEXT		StringRange
input.name.20	TEXT		StringRange
input.name.21	TEXT		StringRange
input.name.22	TEXT		StringRange
input.name.23	TEXT		StringRange

name	*type*	*flags*	*range*
input.name.24	TEXT		StringRange
input.name.25	TEXT		StringRange
input.name.26	TEXT		StringRange
input.name.27	TEXT		StringRange
input.name.28	TEXT		StringRange
input.name.29	TEXT		StringRange
input.name.30	TEXT		StringRange
input.name.31	TEXT		StringRange
input.name.32	TEXT		StringRange
input.name.33	TEXT		StringRange
input.name.34	TEXT		StringRange
input.name.35	TEXT		StringRange
input.name.36	TEXT		StringRange
input.name.37	TEXT		StringRange
input.name.38	TEXT		StringRange
input.name.39	TEXT		StringRange
input.name.40	TEXT		StringRange
input.name.41	TEXT		StringRange
input.name.42	TEXT		StringRange
input.name.43	TEXT		StringRange
input.name.44	TEXT		StringRange
input.name.45	TEXT		StringRange
input.name.46	TEXT		StringRange
input.name.47	TEXT		StringRange
input.name.48	TEXT		StringRange
input.name.49	TEXT		StringRange

name	*type*	*flags*	*range*
input.name.50	TEXT		StringRange
input.name.51	TEXT		StringRange
input.name.52	TEXT		StringRange
input.name.53	TEXT		StringRange
input.name.54	TEXT		StringRange
input.name.55	TEXT		StringRange
input.name.56	TEXT		StringRange
input.name.57	TEXT		StringRange
input.name.58	TEXT		StringRange
input.name.59	TEXT		StringRange
input.name.60	TEXT		StringRange
input.name.61	TEXT		StringRange
input.name.62	TEXT		StringRange
input.name.63	TEXT		StringRange
input.name.64	TEXT		StringRange
input.level.01	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.02	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.03	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.04	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.05	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.06	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.07	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.08	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.09	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.10	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.11	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.level.12	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.13	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.14	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.15	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.16	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.17	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.18	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.19	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.20	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.21	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.22	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.23	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.24	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.25	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.26	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.27	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.28	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.29	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.30	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.31	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.32	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.33	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.34	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.35	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.36	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.37	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.level.38	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.39	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.40	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.41	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.42	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.43	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.44	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.45	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.46	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.47	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.48	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.49	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.50	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.51	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.52	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.53	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.54	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.55	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.56	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.57	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.58	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.59	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.60	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.61	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.62	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.level.63	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.level.64	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.state.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.11	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.12	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.13	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.14	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.15	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.16	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.17	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.18	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.19	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.20	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.21	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.22	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.23	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.24	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.25	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.state.26	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.27	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.28	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.29	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.30	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.31	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.32	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.33	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.34	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.35	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.36	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.37	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.38	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.39	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.40	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.41	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.42	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.43	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.44	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.45	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.46	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.47	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.48	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.49	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.50	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.state.51	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.52	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.53	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.54	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.55	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.56	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.57	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.58	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.59	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.60	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.61	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.62	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.63	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.state.64	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.11	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.12	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.card.13	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.14	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.15	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.16	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.17	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.18	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.19	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.20	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.21	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.22	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.23	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.24	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.25	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.26	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.27	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.28	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.29	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.30	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.31	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.32	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.33	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.34	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.35	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.36	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.37	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.38	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.39	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.card.40	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.41	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.42	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.43	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.44	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.45	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.46	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.47	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.48	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.49	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.50	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.51	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.52	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.53	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.54	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.55	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.56	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.57	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.58	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.59	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.60	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.61	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.62	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.63	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.card.64	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.opt.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.11	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.12	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.13	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.14	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.15	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.16	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.17	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.18	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.19	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.20	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.21	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.22	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.23	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.24	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.25	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.26	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.27	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.opt.28	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.29	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.30	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.31	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.32	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.33	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.34	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.35	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.36	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.37	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.38	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.39	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.40	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.41	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.42	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.43	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.44	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.45	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.46	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.47	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.48	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.49	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.50	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.51	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.52	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
input.opt.53	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.54	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.55	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.56	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.57	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.58	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.59	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.60	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.61	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.62	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.63	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
input.opt.64	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
output.name.01	TEXT		StringRange
output.name.02	TEXT		StringRange
output.name.03	TEXT		StringRange
output.name.04	TEXT		StringRange
output.name.05	TEXT		StringRange
output.name.06	TEXT		StringRange
output.name.07	TEXT		StringRange
output.name.08	TEXT		StringRange
output.name.09	TEXT		StringRange
output.name.10	TEXT		StringRange
output.name.11	TEXT		StringRange
output.name.12	TEXT		StringRange
output.name.13	TEXT		StringRange
output.name.14	TEXT		StringRange

name	*type*	*flags*	*range*
output.name.15	TEXT		StringRange
output.name.16	TEXT		StringRange
output.name.17	TEXT		StringRange
output.name.18	TEXT		StringRange
output.name.19	TEXT		StringRange
output.name.20	TEXT		StringRange
output.name.21	TEXT		StringRange
output.name.22	TEXT		StringRange
output.name.23	TEXT		StringRange
output.name.24	TEXT		StringRange
output.name.25	TEXT		StringRange
output.name.26	TEXT		StringRange
output.name.27	TEXT		StringRange
output.name.28	TEXT		StringRange
output.name.29	TEXT		StringRange
output.name.30	TEXT		StringRange
output.name.31	TEXT		StringRange
output.name.32	TEXT		StringRange
output.name.33	TEXT		StringRange
output.name.34	TEXT		StringRange
output.name.35	TEXT		StringRange
output.name.36	TEXT		StringRange
output.name.37	TEXT		StringRange
output.name.38	TEXT		StringRange
output.name.39	TEXT		StringRange
output.name.40	TEXT		StringRange
output.name.41	TEXT		StringRange

name	*type*	*flags*	*range*
output.name.42	TEXT		StringRange
output.name.43	TEXT		StringRange
output.name.44	TEXT		StringRange
output.name.45	TEXT		StringRange
output.name.46	TEXT		StringRange
output.name.47	TEXT		StringRange
output.name.48	TEXT		StringRange
output.name.49	TEXT		StringRange
output.name.50	TEXT		StringRange
output.name.51	TEXT		StringRange
output.name.52	TEXT		StringRange
output.name.53	TEXT		StringRange
output.name.54	TEXT		StringRange
output.name.55	TEXT		StringRange
output.name.56	TEXT		StringRange
output.name.57	TEXT		StringRange
output.name.58	TEXT		StringRange
output.name.59	TEXT		StringRange
output.name.60	TEXT		StringRange
output.name.61	TEXT		StringRange
output.name.62	TEXT		StringRange
output.name.63	TEXT		StringRange
output.name.64	TEXT		StringRange
output.name.65	TEXT		StringRange
id.out1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
id.out3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out25	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out26	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out27	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
id.out28	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out29	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out30	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out31	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out32	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out33	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out34	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out35	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out36	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out37	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out38	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out39	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out40	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out41	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out42	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out43	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out44	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out45	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out46	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out47	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out48	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out49	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out50	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out51	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out52	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out53	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out54	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
id.out55	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out56	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out57	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out58	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out59	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out60	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out61	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out62	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out63	INTEGER	R/O	IntegerRange R/O (0 .. 0)
id.out64	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reInit	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Int status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ext status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.157 DEV-Matrix-1996

This page describes the device driver and the device window for a DEV 1996 L-Band Matrix with up to 32 inputs and outputs. If the number of interfaces changes it requires restarting of the M&C service, because the interface number is only read one time at startup.

Alias names are stored locally in satnms only as they cannot be read from the unit. The driver should also work with DEV 1995 CATV matrices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port Switching by Alias Name.
- --- Port Switching by Number.
- --- Edit the Input and Output Alias Names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DEV 1986 matrix.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Delays the read back of a crosspoint setting actually after it has been set by the driver to at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
ctrlMode	Sets the control mode Local or Remote.
switchingMode	Sets the Switching Mode of the Matrix. Please be aware that the driver will only work in Table mode.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.ctrlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.switchingMode	CHOICE		EnumRange (TABLE STRING)
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.17	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.19	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.21	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.22	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.23	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.24	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.25	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.26	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.27	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.28	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.29	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.30	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.31	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.32	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
byname.01	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.02	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.03	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.04	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.05	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.06	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.07	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.08	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.09	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.10	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.11	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.12	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.13	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.14	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.15	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.16	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.17	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.18	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.19	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.20	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.21	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.22	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.23	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.24	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.25	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.26	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.27	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.28	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)

name	*type*	*flags*	*range*
byname.29	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.30	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.31	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
byname.32	CHOICE		EnumRange (Terminate Input 1 Input 2 Input 3 Input 4 Input 5 Input 6 Input 7 Input 8 Input 9 Input 10 Input 11 Input 12 Input 13 Input 14 Input 15 Input 16 Input 17 Input 18 Input 19 Input 20 Input 21 Input 22 Input 23 Input 24 Input 25 Input 26 Input 27 Input 28 Input 29 Input 30 Input 31 Input 32)
input.name.01	TEXT	SAVE	StringRange
input.name.02	TEXT	SAVE	StringRange
input.name.03	TEXT	SAVE	StringRange
input.name.04	TEXT	SAVE	StringRange
input.name.05	TEXT	SAVE	StringRange
input.name.06	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
input.name.07	TEXT	SAVE	StringRange
input.name.08	TEXT	SAVE	StringRange
input.name.09	TEXT	SAVE	StringRange
input.name.10	TEXT	SAVE	StringRange
input.name.11	TEXT	SAVE	StringRange
input.name.12	TEXT	SAVE	StringRange
input.name.13	TEXT	SAVE	StringRange
input.name.14	TEXT	SAVE	StringRange
input.name.15	TEXT	SAVE	StringRange
input.name.16	TEXT	SAVE	StringRange
input.name.17	TEXT	SAVE	StringRange
input.name.18	TEXT	SAVE	StringRange
input.name.19	TEXT	SAVE	StringRange
input.name.20	TEXT	SAVE	StringRange
input.name.21	TEXT	SAVE	StringRange
input.name.22	TEXT	SAVE	StringRange
input.name.23	TEXT	SAVE	StringRange
input.name.24	TEXT	SAVE	StringRange
input.name.25	TEXT	SAVE	StringRange
input.name.26	TEXT	SAVE	StringRange
input.name.27	TEXT	SAVE	StringRange
input.name.28	TEXT	SAVE	StringRange
input.name.29	TEXT	SAVE	StringRange
input.name.30	TEXT	SAVE	StringRange
input.name.31	TEXT	SAVE	StringRange
input.name.32	TEXT	SAVE	StringRange
output.name.01	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
output.name.02	TEXT	SAVE	StringRange
output.name.03	TEXT	SAVE	StringRange
output.name.04	TEXT	SAVE	StringRange
output.name.05	TEXT	SAVE	StringRange
output.name.06	TEXT	SAVE	StringRange
output.name.07	TEXT	SAVE	StringRange
output.name.08	TEXT	SAVE	StringRange
output.name.09	TEXT	SAVE	StringRange
output.name.10	TEXT	SAVE	StringRange
output.name.11	TEXT	SAVE	StringRange
output.name.12	TEXT	SAVE	StringRange
output.name.13	TEXT	SAVE	StringRange
output.name.14	TEXT	SAVE	StringRange
output.name.15	TEXT	SAVE	StringRange
output.name.16	TEXT	SAVE	StringRange
output.name.17	TEXT	SAVE	StringRange
output.name.18	TEXT	SAVE	StringRange
output.name.19	TEXT	SAVE	StringRange
output.name.20	TEXT	SAVE	StringRange
output.name.21	TEXT	SAVE	StringRange
output.name.22	TEXT	SAVE	StringRange
output.name.23	TEXT	SAVE	StringRange
output.name.24	TEXT	SAVE	StringRange
output.name.25	TEXT	SAVE	StringRange
output.name.26	TEXT	SAVE	StringRange
output.name.27	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
output.name.28	TEXT	SAVE	StringRange
output.name.29	TEXT	SAVE	StringRange
output.name.30	TEXT	SAVE	StringRange
output.name.31	TEXT	SAVE	StringRange
output.name.32	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (String Mode)

9.158 DEV-Switch-2-To-1

This page describes the **DEV-Switch-2-To-1** device driver and its device window. The**DEV-Switch-2-To-1** device driver monitors and controls one 2:1 switch module card.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the DEV chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:

- DEV 13-0369 1:2 Optical Switch Module in a DEV 1953 chassis
- SW version I.01

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info c.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
- switchNumber --- Selects the switch number to be controlled.
- useId --- Defines is the port indexes are calculated automatically based on the number of available modules. Using AUTO should work on all frames using a common card type in all slots. If AUTO selection is not matching use the DEV Frame driver overview to determine port indexes.
- idInputA --- Sets the port index of input A in Manual Mode.
- idInputB --- Sets the port index of input B in Manual Mode.
- idOutput --- Sets the port index of the output in Manual Mode.
- rfLimit --- Defines the rf (switchover) limit in dBm.
- monitoring --- En/Disable power monitoring if applicable.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.module	TEXT	R/O	StringRange R/O
info.hasRfSense	CHOICE	R/O	EnumRange R/O (YES NO)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.switchNumber	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.useId	CHOICE	SAVE SETUP	EnumRange (AUTO MANUAL)
config.idInputA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.idInputB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.idOutput	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.rfLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.monitoring	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.ctrlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE AUTO)
input.a	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.b	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
position	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Card)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input A)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input B)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Surveillance)

9.159 Dantherm-ComboCooling1000

This page describes the monitoring only device driver and the device window for the Dantherm Combo Cooling 10000/1000 air conditioning controller device using Modbus protocol.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The measurement values and device states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

The Dantherm unit is controlled through Modbus/TCP. The device IP address must be set as an interface name like with serial devices controlled through a network terminal adapter:

"aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.product	TEXT	R/O	StringRange R/O
fan.1.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.2.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.3.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.1.drive	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.2.drive	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.3.drive	INTEGER	R/O	IntegerRange R/O (0 .. 0)
room.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ambient.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
supply.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
condensor.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
board.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
supply.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
system.state	TEXT	R/O	StringRange R/O
fan.1.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
fan.2.state	TEXT	R/O	StringRange R/O
fan.3.state	TEXT	R/O	StringRange R/O
damper.1.state	TEXT	R/O	StringRange R/O
damper.2.state	TEXT	R/O	StringRange R/O
compr.state	TEXT	R/O	StringRange R/O
ac.1.state	TEXT	R/O	StringRange R/O
ac.2.state	TEXT	R/O	StringRange R/O
heater.state	TEXT	R/O	StringRange R/O
damper.1.position	INTEGER	R/O	IntegerRange R/O (0 .. 0)
damper.2.position	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Alarm 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Alarm 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temp limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Onboard sensor)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Room sensor)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ambient sensor)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Supply sensor)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Condensor sensor)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Digital 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Digital 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Door switch)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Filter inout)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Phase monitor)

9.160 Dantherm-TKS3000A

This page describes the device driver and the device window for the Dantherm TKS3000A weather station.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
indoorTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
outdoorTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fanSpeed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.acMode	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.currentDetError	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.acExercise	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.timerActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.totalStopActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.seriousConfigError	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.eepromError	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.f1Alarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.f2Alarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.fireAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.filterAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lowTempAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.highTempAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.tcFunctionInstalled	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ac1UnitInstalled	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.slaveOperation	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.motorizedDampers	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.internalHeater	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.currentDetectActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ac2UnitInstalled	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.heaterActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ac1UnitActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.exhaustDamperOn	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
flags.ac2UnitActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.outputFilterAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.outputFanAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.outputTcFuncActive	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.outputLowTempAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.outputHighTempAlarm	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Current detection)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.04	ALARM	R/O	AlarmFlagRange R/O (F1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (F2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fire)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Filter)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Low temp)
faults.09	ALARM	R/O	AlarmFlagRange R/O (High temp)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output filter Alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output fan)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output low temp)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output high temp)

9.161 De-Icing-Buttons

The *De-Icing-Buttons* logical device controls the de-icing circuits of one or more antennas using a group of buttons. The idea behind this device is not to bind the antenna heating directly to the recognition of the "snow" condition, but to give the operator the opportunity to react to this condition in a meaningful and convenient way. Controlling the antenna heater works in the following way:

1. The meteorological station detects precipitation and / or low temperature. It may be a good idea to introduce some minutes delay there to verify it is really snowing (this must happen outside the De-Icing-Buttons device, e.g. by means of a Delayed-Gate-3 device.
2. After this delay, the device is showing *Advance warning, Deicing Auto ON* for 'preDelay' minutes (configurable, 2 minutes default). During this time the operator can decide to use one of the following modes:
   - AUTO-mode: The deicing will be turned ON after the delay of 'preDelay' minutes
   - MANUAL-ON: The deicing will be turned ON for 'manualTime' minutes (configurable, uses s one hour default). After this time the device will return to automatic mode
   - MANUAL-OFF: The deicing will be turned OFF for 'manualTime' minutes. After this time the device will return to automatic mode
3. By pressing the button "RESTART" the timer for MANUAL-ON or -OFF will be restarted. The same happens, if the MANUAL-ON or MANUAL-OFF button is clicked again.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the actual state and the buttons of the device
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
detectorId	Defines the ID of the parameter which signals that the "snow" condition is active. The parameter must be of type BOOL and must signal the snow condition as 'true'.
heaterId	Defines the ID of the parameter which switches the heating of this controller actually ON/OFF. The parameter is expected to be a CHOICE type with the selection "OFF" and "ON".
preDelay	The delay time (minutes) the device waits after snow detection until AUTO mode is started.
manualTime	The duration (minutes) of the MANUAL-ON or MANUAL-OFF states.

Configuring the De-Icing-Buttons togeter with a De-Icing-Multiplex Device

To use De-Icing-Buttons device as the input of aDe-Icing-Multiplex device, you have to set the

'heatingRequestedId.#' configuration parameters of all used channels in the De-Icing-Multiplex device to the 'heating' status parameter of the De-Icing-Buttons device. The 'heaterId' configuration parameter of the De-Icing-Buttons device remains empty in this case.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.detectorId	TEXT	SAVE SETUP	StringRange
config.heaterId	TEXT	SAVE SETUP	StringRange
config.preDelay	INTEGER	SAVE SETUP	IntegerRange (1 .. 10)
config.manualTime	INTEGER	SAVE SETUP	IntegerRange (5 .. 180)
state	TEXT	R/O	StringRange R/O
remaining	TEXT	R/O	StringRange R/O
heating	BOOLEAN	R/O	BooleanRange R/O (true false)
snowDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
advanceWarning	BOOLEAN	R/O	BooleanRange R/O (true false)
auto	TEXT	NOPRESET	StringRange
manualOn	TEXT	NOPRESET	StringRange
manualOff	TEXT	NOPRESET	StringRange
restart	TEXT	NOPRESET	StringRange

9.162 De-Icing-Multiplex

The *De-Icing-Multiplex* logical device controls up to eight antenna de-icing controllers to ensure that only one de-icing controller has it's heaters switched on at a time. The *De-Icing-Multiplex* device is used to reduce the maximum current draw by implementing this restriction.

The device requires that the de-icing controllers state a "I want to heat" flag with a BOOL parameter. The de-icing controls also must provide an OFF/ON CHOICE parameter to switch the heater actually off or on.

The *De-Icing-Multiplex* logical device implements a round-robin scheme to switch on the heaters (at those controllers which have the "request" flag set) one after the other after the "maxHeatingTime" timer runs down. The scheme preemptively releases the heating state, if a controller drops it's request flag.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the actual state of the multiplexer
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
heatingRequestedId.*	Defines the ID of the parameter which signals that this de-icing controller now requests heating. The parameter is expected to be a BOOL type.
heatingCommandId.*	Defines the ID of the parameter which switches the heating of this controller actually ON/OFF. The parameter is expected to be a CHOICE type with the selection "OFF" and "ON".
maxHeatingTime	Defines maximum time (minutes), the multiplexer switches on the heating at one particular controller. This is the round-robin time, after which the multiplexer proceeds to the next controller if more than one controller requests heating.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.heatingRequestedId.1	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.heatingRequestedId.2	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.3	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.4	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.5	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.6	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.7	TEXT	SAVE SETUP	StringRange
config.heatingRequestedId.8	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.1	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.2	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.3	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.4	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.5	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.6	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.7	TEXT	SAVE SETUP	StringRange
config.heatingCommandId.8	TEXT	SAVE SETUP	StringRange
config.maxHeatingTime	INTEGER	SAVE SETUP	IntegerRange (1 .. 30)

name	*type*	*flags*	*range*
heatingRequested.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.4	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.5	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.6	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.7	BOOLEAN	R/O	BooleanRange R/O (true false)
heatingRequested.8	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.4	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.5	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.6	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.7	BOOLEAN	R/O	BooleanRange R/O (true false)
heating.8	BOOLEAN	R/O	BooleanRange R/O (true false)

9.163 Dektec-DTA2137

This page describes the device driver and the device window for the Dektec DTA-2137 dual DVB-S2 demodulator card. The sat-nms software operates this card as a receive quality measurement device only. No demuxer, or video/audio decoder is included with the software used for this card. However, the demodulated transport streams are output to the ASI connectors of the card.

ATTENTION

To operate one or more DTA-2137 cards with the sat-nms software, it is necessary to install additional proprietary software from Dektec on the VLC/M&C computer. For details please read the section *Installation* at the end of this page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the receive settings like frequency and symbol rate.
- --- Shows the actual demodulator values including the figures of receive level and quality.
- --- Shows the transport stream info evaluated by the device driver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to this device.

parameter	***description***
address	Leave this field empty, the protocol address is not used by the device driver. Each individual demodulator unit is addressed by the interface address specified in the device configuration screen (See 'installation' at the end of this page).
22kHz	Makes the device apply a 22 kHz tone to the LNB power when set to ON.

parameter	***description***
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Selects if the device shall apply LNB power to the input.
lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
lnbPolSwitch	If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Installation

The DTA-2137 is a dual DVB-S2 demodulator, hence there are two instances of the driver to be configured for each card.

When configuring a Dektec-DTA2137 device in the device configuration screen, you have to specify the card's PCI express bus/slot number instead of the device's interface name. The address consists of 5 digits *"xx:yy.z"*, where:

- *xx* is the PCI bus number of the card (hexadecimal, as reported by lspci)
- *yy* is the PCI slot number of the card (hexadecimal, as reported by lspci)
- *z* is 1 or 2, selecting the demodulator 1 or 2 on the card.

Example: *"09:00.1"* specifies the first demodulator on the card located at PCI bus 09, slot 00.

Important Note : The DTA-2137 requires some proprietary software from Dektec running on the VLC/M&C to make the sat-nms software communicate with the demodulators. Please

contact SatService GmbH if you intend to use Dektec DTA-2137 cards on a system which is not prepared for this.

Transport stream info

The device driver reads and evaluates the transport stream received with a properly tuned card. The transport stream information remains visible even if the card loses the input signal. The transport stream information my be reset by clicking the "RESET TS INFO" button or by tuning to a new frequency.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.cardid	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SAVE SETUP	EnumRange (13V 14V 18V 19V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
dvbmode	CHOICE	SAVE	EnumRange (DVBS DVBS2)
frequency	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
modulation	CHOICE	SAVE	EnumRange (QPSK 8PSK 16APSK 32APSK)
polarization	CHOICE	SAVE	EnumRange (H V)
flags.feclock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rxlock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.preBer	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.postBer	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.preBerLog	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.postBerLog	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dvbmode	TEXT	R/O	StringRange R/O
state.modulation	TEXT	R/O	StringRange R/O
state.fec	TEXT	R/O	StringRange R/O
state.spectrum	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.pilots	TEXT	R/O	StringRange R/O
state.fecframe	TEXT	R/O	StringRange R/O
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ts.networkName	TEXT	R/O	StringRange R/O
ts.services	CHOICE	R/O	EnumRange R/O (NO SERVICES)
ts.syncLossCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ts.syncErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ts.packetErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ts.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ts.pat2ErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ts.pmt2ErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ts.reset	TEXT	NOPRESET	StringRange
ts.resetFaults	TEXT	NOPRESET	StringRange
ts.inSync	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
reset	TEXT	NOPRESET	StringRange
startupDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.l	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.s1	TEXT	R/O	StringRange R/O
internal.s2	TEXT	R/O	StringRange R/O
internal.ber1	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.ber2	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.syncLossCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.syncErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.packetErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pat2ErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pmt2ErrCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (FEC Lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RX Lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sync loss count)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Sync error count)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Packet error count)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PAT error 2 count)
faults.07	ALARM	R/O	AlarmFlagRange R/O (PMT error 2 count)

9.164 Dektec-DTA2137v4

This page describes the device driver and the device window for the Dektec DTA-2137 dual DVB-S2 demodulator card. The sat-nms software operates this card as a receive quality measurement device only. No demuxer, or video/audio decoder is included with the software used for this card. However, the demodulated transport streams are output to the ASI connectors of the card.

ATTENTION

This version works with DTAPI 4.x

To operate one or more DTA-2137 cards with the sat-nms software, it is necessary to install additional proprietary software from Dektec on the VLC/M&C computer. For details please read the section *Installation* at the end of this page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the receive settings like frequency and symbol rate.

- --- Shows the actual demodulator values including the figures of receive level and quality.
- --- Shows the transport stream info evaluated by the device driver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to this device.

parameter	***description***
address	Leave this field empty, the protocol address is not used by the device driver. Each individual demodulator unit is addressed by the interface address specified in the device configuration screen (See 'installation' at the end of this page).
22kHz	Makes the device apply a 22 kHz tone to the LNB power when set to ON.
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Selects if the device shall apply LNB power to the input.
lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
lnbPolSwitch	If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Installation

The DTA-2137 is a dual DVB-S2 demodulator, hence there are two instances of the driver to be configured for each card.

When configuring a Dektec-DTA2137 device in the device configuration screen, you have to specify the card's PCI express bus/slot number instead of the device's interface name. The address consists of 5 digits *"xx:yy.z"*, where:

- *xx* is the PCI bus number of the card (hexadecimal, as reported by lspci)
- *yy* is the PCI slot number of the card (hexadecimal, as reported by lspci)
- *z* is 1 or 2, selecting the demodulator 1 or 2 on the card.

Example: *"09:00.1"* specifies the first demodulator on the card located at PCI bus 09, slot 00.

Important Note : The DTA-2137 requires some proprietary software from Dektec running on the VLC/M&C to make the sat-nms software communicate with the demodulators. Please contact SatService GmbH if you intend to use Dektec DTA-2137 cards on a system which is not prepared for this.

Transport stream info

The device driver reads and evaluates the transport stream received with a properly tuned card. The transport stream information remains visible even if the card loses the input signal. The transport stream information my be reset by clicking the "RESET TS INFO" button or by tuning to a new frequency.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.cardid	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SAVE SETUP	EnumRange (13V 14V 18V 19V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
dvbmode	CHOICE	SAVE	EnumRange (DVBS DVBS2)
frequency	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
modulation	CHOICE	SAVE	EnumRange (AUTO QPSK 8PSK 16APSK 32APSK)
fec	CHOICE	SAVE	EnumRange (AUTO 1/2 2/3 3/4 5/6 6/7 7/8 8/9 1/4 1/3 2/5 3/5 4/5 9/10)
polarization	CHOICE	SAVE	EnumRange (H V)

name	*type*	*flags*	*range*
flags.feclock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rxlock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.preBer	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.postBer	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.preBerLog	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.postBerLog	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dvbmode	TEXT	R/O	StringRange R/O
state.modulation	TEXT	R/O	StringRange R/O
state.fec	TEXT	R/O	StringRange R/O
state.spectrum	TEXT	R/O	StringRange R/O
state.pilots	TEXT	R/O	StringRange R/O
state.fecframe	TEXT	R/O	StringRange R/O
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.f	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.l	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.s1	TEXT	R/O	StringRange R/O
internal.s2	TEXT	R/O	StringRange R/O
internal.ber1	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.ber2	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (FEC Lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RX Lock)

9.165 Delay-Line-Mirror

The Delay-Line-Mirror logical device complements the Delay-Line device for applications in a N:1 protection setup. The Delay-Line-Mirror duplicates the transmit power setting of the redundant chain to a parameter called 'output' which can be mirrored from the Delay-Line devices in the operative chains. For a complete description how Delay-Line devices work and why to use them to mirror the transmit power in a N:1 protection setup read the description of the Delay-Line device.

To configure a Protection-Switch-N-To-1 device with Delay-Line devices in the operative chains and a Delay-Line-Mirror in the spare chain, setup the devices as follows:

- For each operative setup a Delay-Line device and make it watch the output power of this chain. If applicable, set the offset parameter each Delay-Line to compensate for the signal path attenuation differences occuring when a HPA is replaced by the redundant one.
- For the redundant chain setup a Delay-Line-Mirror device which replicates the spare chain output power.
- If the Protection-Switch-N-To-1 device uses the automatic enumeration / indexing of device names, name the devices according to this scheme (eg. DLY-1, DLY-2 .. DLY-R).
- Set the Delay-Line/Delay-Line-Mirror devices as the Level device (levelPrefix) and choose DELAY-LINE as Level type (levelSetType) in the Protection-Switch-N-To-1 device.

The Protection-Switch-N-To-1 device will mirror the output power which was active some seconds before to the spare chain in case of a switchover.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual output power of the assigned, apre chain.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Delay-Line-Mirror device with the device that sets the output power of the spare chain.

parameter	***description***
pwrSetId	Enter the ID of the parameter which sets the output power of the chain.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.pwrSetId	TEXT	SAVE SETUP	StringRange
output	FLOAT		DoubleRange (0.0 .. 0.0)

9.166 Delay-Line

The Delay-Line logical device watches an arbitrary input parameter (may be a read-write or a read-only parameter) and passes this value through a delay line of configurable length. The output value of the device represents the value the input parameter had at 'delay' seconds before.

The delay line is build as a shift register with*n* stages, where *n* is the number of seconds set as the delay time. Every second the actual input value is put into the shift register, the oldest value from the register is dropped out to the output value of the device.

The one second clock for the shift register is derived from the cycle time of the device thread, hence the device must be configured in its own 'interface' (device thread) with 1000 milliseconds idle time.

The Delay-Line works with arbitrary parameter types, the 'input' and 'output' variables of the device change their type and range definition to that of the watched input parameter. When using numeric values, an offset may be added to the output value.

Using the Delay-Line with a N:1 protection switch

A N:1 protection switch usually mirrors the output power of the faulty chain to the spare one during switchover. With power amplifiers which regulate the output power by themselves, the (too low) output power itself may be the reason for the switchover. In such a case it makes no sense to mirror the power to the spare amplifier because this will also fail due to the wrong value.

Using the Delay-Line device can overcome this problem: For each operational transmit chain a Delay-Line device is configured which watches the output power of the HPA in this chain. The offset value of each Delay-Line may be set to compensate for the signal path attenuation difference between operational and spare HPA. The output of each Delay-Line device contains the output power of the chain which was present e.g. 10 seconds ago. Together with the offset value this is exactly the power which must be set at the spare HPA in case of a switchover.

For the redundant chain, a Delay-Line-Mirror device gets configured to set the mirrored output

power at the spare HPA. The Protection-Switch-N-To-1 device is set to mirror the Delay-Line / Delay-Line-Mirror devices as an Level device of the transmit chains.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual input output values of the device, permits to set the 'delay' and 'offset' values.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Delay-Line device with the device that delivers the input signal.

parameter	***description***
inputId	Enter the ID of the parameter the device shall watch as its input value.

Remarks

- Changing the delay time resets the shift register. The output will read the input value which was valid at the time when the delay time was changed for the next 'delay' seconds. Then it reflects the changes recorded for the input value after this moment.
- Changing the 'inputId' configuration parameter resets the shift register as well and also changes the data type definition of the 'input' and 'output' parameters. The 'offset' parameter is only enabled, if the data type of the watched parameter is INTEGER or FLOAT. After changing the 'inputId' configuration parameter, the device window of the Delay-Line should be closed and opened again to make the window adopt all changes.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputId	TEXT	SAVE SETUP	StringRange

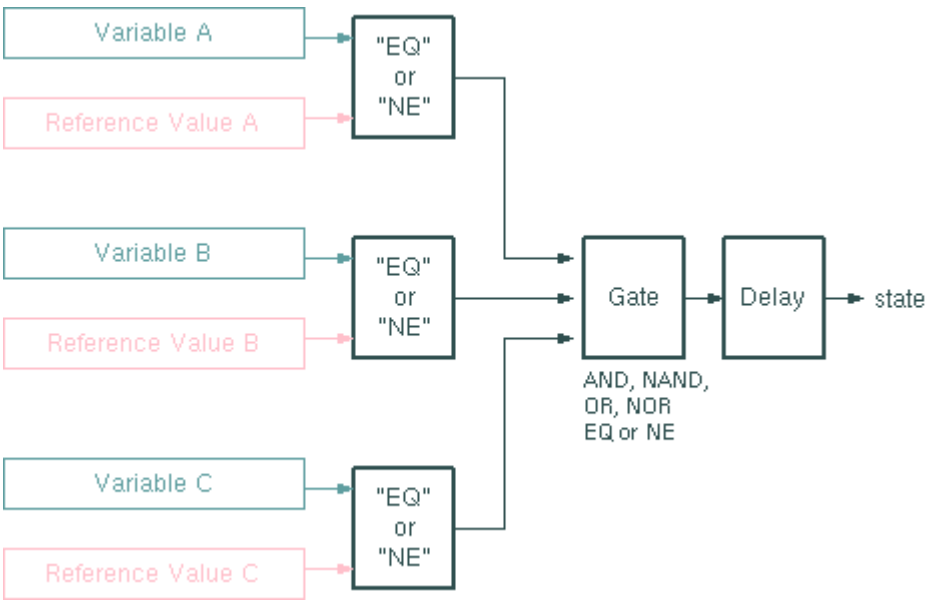
name	*type*	*flags*	*range*
input	TEXT	R/O	StringRange R/O
output	TEXT		StringRange
delay	INTEGER	SAVE	IntegerRange (0 .. 300)
offset	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)

9.167 Delayed-Gate-3

The Delayed-Gate-3 device provides a 3-input logical gate which can be used to merge status variables in a user programmed way. Delayed-Gate-3 works much like the Logical-Gate-3 device but additionally has a delay unit processing the signal downstream. The delay makes the the output become true only if the gate condition becomes true for at least the configured delay time

The gate may be configured to work as a AND, NAND, OR, NOR, NE (XOR) or EQ gate. It listens to two input variables and compares these to user supplied reference values. the logical result of the comparisons gets fed into the gate. The gate's output state is passed though the delay unit and then distributed as a boolean state variable named "state".

The temporal resolution of the delay unit is determined by the cycle time of the device thread ("PORT") the Delayed-Gate-3 is configured in. Thus, if you plan to use short delay times (0.1 to 1.5 seconds), it is recommended to configure the Delayed-Gate-3 as the only device in a device thread using a 100 msecs idle time. This ensures, that the delay is executed reliably with the configured time.



Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Logical-Gate device with the devices that deliver the gate's input signals.

parameter	***description***
input-A.Id	Enter the message identifier here of the parameter the gate shall listen as input-A, e.g. "SWITCH-1. position".
input-A.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-A.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
gateType	Select the type of logical gate.
input-B.Id	Enter the message identifier here of the parameter the gate shall listen as input-B, e.g. "SWITCH-2. position".
input-B.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-B.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
input-C.Id	Enter the message identifier here of the parameter the gate shall listen as input-C, e.g. "SWITCH-3. position".

parameter	*description*
input-C.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-C.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
setFaultOnTrue	If set to YES, the device sets the device summary fault and a fault flag **faults.01** if the gate output is true.
delayTime	The delay time to apply (seconds)

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.gateType	CHOICE	SAVE SETUP	EnumRange (AND NAND OR NOR EQ NE)
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-A.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-A.reference	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.input-B.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-B.reference	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-C.Id	TEXT	SAVE SETUP	StringRange
config.input-C.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-C.reference	TEXT	SAVE SETUP	StringRange
config.setFaultOnTrue	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.delayTime	FLOAT	SAVE SETUP	DoubleRange (0.1 .. 3600.0)
state	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Gate output)

9.168 Device-Copy

The Device-Copy logical device copies all settings from one device to another when the operator commands this. The Device-Copy behaves like a manual operated Device-Mirror.

The main difference to the Device-Mirror device is, that the Device-Copy device leaves the slave device un-touched until the operator presses the 'copy' button. Then the device one by one compares the settings of the two devices. Setting which differ between the two devices are set at device 2 to the value set at device 1.

The Device-Copy generally is used for semi-automatic backup / redundancy switching systems where the operator has to decide when to mirror the settings to the redundant device.

The device provides it's own device window with the "copy" button which triggers the parameter copy. In most cases however you will want to trigger the copy from a button at a user screen. To configure this, place a "ParameterButton" element at the screen which writes an arbitrary string value to the 'copy' variable of the Device-Copy device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Provides the copy button.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Device-Copy device with the devices it shall mirror.

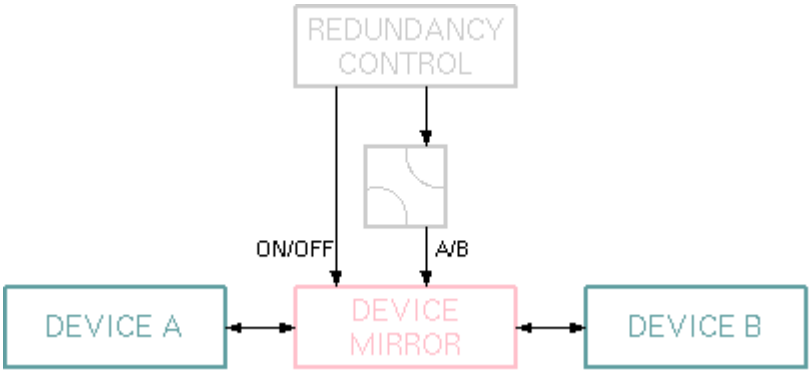
parameter	*description*
device1Name	The name of device 1 (the master device).
device2Name	The name of device 2 (the slave device).

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.device1Name	TEXT	SAVE SETUP	StringRange
config.device2Name	TEXT	SAVE SETUP	StringRange
copy	TEXT	NOPRESET	StringRange

9.169 Device-Mirror

The Device-Mirror logical device ties two 1:1 redundant devices together. Any time a setting is changed at the primary device, the Device-Mirror copies this setting to the spare device. Both devices must be of the same type. The Device-Mirror itself provides no redundancy control or switching, it is intended to be used as an extension of an existing redundancy switching unit.



The Device-Mirror works symmetrically, there is no fixed definition of a primary and a spare device. To determine it's working direction, the Device-Mirror monitors a switch position. This switch position may be an arbitrary M&C parameter, hence a hardware switch may be monitored as well as the output state of a software based redundancy control. Accordingly, the redundancy controller may be both, a sat-nms logical device or an external hardware unit.

Mirroring the device settings usually only is desired while the redundancy / protection switch is activated. The Device-Mirror monitors a second parameter, which enables mirroring. This usually is the ON/OFF setting of the redundancy controller. While the Device-Mirror is disabled by this parameter, no mirroring of settings takes place. However, the Device-Mirror monitors all settings of both devices and remembers them in an internal memory. Once the redundancy switch is enabled, the Device-Mirror controls the spare device to match the settings read from the active one.

The Device-Mirror controls most of the operative parameters of the spare device. The list of parameters which are mirrored is the same set which is stored into device presets of this device type. Configuration parameters usually are excluded as well as "event trigger" parameters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Device-Mirror device with the devices it shall mirror.

parameter	***description***
enableId	Enter the message identifier here of the parameter the Device-Mirror shall monitor in order to enable / disable the mirroring function. (e.g. "PROTECT.enable")
enableVal	This is the message contents, the Device-Mirror expects to enable the mirroring. (e.g. "ON")
directionId	Enter the message identifier here of the switch position which determines the direction of the mirroring function. (e.g. "S2.position")
direction1To2Val	This is the message contents, the Device-Mirror expects to mirror from device 1 to device 2. (e.g. "A")

parameter	*description*
device1Name	The name of device 1 to mirror.
device2Name	The name of device 2 to mirror.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.enableId	TEXT	SAVE SETUP	StringRange
config.enableVal	TEXT	SAVE SETUP	StringRange
config.directionId	TEXT	SAVE SETUP	StringRange
config.direction1To2Val	TEXT	SAVE SETUP	StringRange
config.device1Name	TEXT	SAVE SETUP	StringRange
config.device2Name	TEXT	SAVE SETUP	StringRange

9.170 Device-Switcher

The Device-Switcher logical device lets you access one of two identical physical devices through a common device name. An 'A' / 'B' selection parameter lets you switch between the two devices.

The Device-Switcher device mimics almost completely the actually selected device. Merely the configuration parameters of the device are not passed through, the Device-Switcher always shows its own configuration parameters instead of the parameters of the device it actually mimics.

Beside selecting the device to mimic by the 'A' / 'B' configuration parameter, the Device-Switcher offers a function to watch a given parameter (e.g. a protection switch location) and to select the device depending on the value of this parameter.

Device window pages

The Device-Switcher itself has no device window pages. It shows the device window of the device it actually replicates.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Device-Switcher device with the devices it shall mimic.

parameter	*description*
deviceA	The name of the device to mimic if 'position' is 'A'
deviceB	The name of the device to mimic if 'position' is 'B'
position	Selects the device to mimic, one of 'A' or 'B'
switchId	If this is not empty, the Device-Switcher will watch the parameter with this ID and select the device to mimic based on the content of this parameter
switchPosA	If the value of the parameter addressed by 'switchId' changes to this value, the Device-Switcher changes to position 'A'
switchPosB	If the value of the parameter addressed by 'switchId' changes to this value, the Device-Switcher changes to position 'B'

Remarks

- Unlike other configuration parameters the parameters 'mode' (OPERATIONAL/OUT-OF-SERVICE) and 'verbose' refer to the device the Device-Switcher actually mimics.
- The Device-Switcher works best with two devices which are identical, i.e. the use the same device driver and have the same options installed. You can use different device types with the Device-Switcher, but you have to be aware that some information like the device screen selected will not change automatically when you swap devices.
- The Device-Switcher does not work with devices using the drivers Gate-Array and Proxy-Device as these tweak parameter definitions in a way that is compatible to the Device-Switcher.
- Device-Switchers may not be nested. This is for the same reason as with the device types above, the Device-Switcher does not have its own parameters, it only forwards message from an to the the selected device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
config.deviceA	TEXT	SAVE SETUP	StringRange
config.deviceB	TEXT	SAVE SETUP	StringRange
config.position	CHOICE	SAVE SETUP	EnumRange (A B)
config.switchId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.switchPosA	TEXT	SAVE SETUP	StringRange
config.switchPosB	TEXT	SAVE SETUP	StringRange

9.171 DoubleD-DDA89

This page describes the device driver and the device window for the DoubleD DDA89 N:1 Redundancy Switch. The DDA89 can handle different vendor up- or downconverters in up to 8:1 redundancy configurations.

It was tested with model DDA89-02 in a 2:1 redundancy

- SW 0.24 from 28.11.2004

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch parameters
- --- The channel parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	*description*
address	The device address (ASCII) as set at the device's front panel. Default address is A.
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
remoteMode	Sets the units access mode REMOTE or LOCAL

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.remoteMode	CHOICE		EnumRange (REMOTE LOCAL)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.location	TEXT	R/O	StringRange R/O
internal.0	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.4	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.5	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.6	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.7	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.8	BOOLEAN	R/O	BooleanRange R/O (true false)
ch0.name	TEXT	R/O	StringRange R/O DIS
ch1.name	TEXT	R/O	StringRange R/O DIS
ch2.name	TEXT	R/O	StringRange R/O DIS
ch3.name	TEXT	R/O	StringRange R/O DIS
ch4.name	TEXT	R/O	StringRange R/O DIS
ch5.name	TEXT	R/O	StringRange R/O DIS
ch6.name	TEXT	R/O	StringRange R/O DIS
ch7.name	TEXT	R/O	StringRange R/O DIS
ch8.name	TEXT	R/O	StringRange R/O DIS
location	CHOICE		EnumRange (STANDBY CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8)
location.1	CHOICE	R/O	EnumRange R/O (A B)
location.2	CHOICE	R/O	EnumRange R/O (A B)
location.3	CHOICE	R/O	EnumRange R/O (A B)
location.4	CHOICE	R/O	EnumRange R/O (A B)
location.5	CHOICE	R/O	EnumRange R/O (A B)
location.6	CHOICE	R/O	EnumRange R/O (A B)
location.7	CHOICE	R/O	EnumRange R/O (A B)
location.8	CHOICE	R/O	EnumRange R/O (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
ch0.on	CHOICE		EnumRange DIS (OFF ON)
ch0.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch0.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch1.on	CHOICE		EnumRange DIS (OFF ON)
ch1.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch2.on	CHOICE		EnumRange DIS (OFF ON)
ch2.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch3.on	CHOICE		EnumRange DIS (OFF ON)
ch3.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch4.on	CHOICE		EnumRange DIS (OFF ON)
ch4.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch5.on	CHOICE		EnumRange DIS (OFF ON)
ch5.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch6.on	CHOICE		EnumRange DIS (OFF ON)
ch6.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch7.on	CHOICE		EnumRange DIS (OFF ON)
ch7.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch7.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch8.on	CHOICE		EnumRange DIS (OFF ON)
ch8.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Serial Comms)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Local)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fault)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fault)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Fault)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Fault)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Fault)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Local)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Local)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Local)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Local)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Local)

name	*type*	*flags*	*range*
faults.25	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Local)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Local)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Local)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Local)

9.172 Downconverter-Protection-Controller

The Downconverter-Protection-Controller logical device builds together with theProtection-Switch-N-To-M device a n:m redundancy solution for satellite downconverters, which supports a pool of backup devices from which one device is taken in case of failure of an operative device. In case of a redundancy switch all device parameters are mirrored to the spare device, so all downconverters in the pool (active and spare ones) must be exactly of the same type.

For n:m redundancy, one Protection-Switch-N-To-M device and for each managed downconverter a Downconverter-Protection-Controller device are configured in the M&C device setup. This distribution across two device types enables a very flexible configuration of the redundancy, so there is no explicit limit on the number of managed downconverters.

The devices are based on the following device configuration:

- The user has a freely definable number of identical devices, some of which act as spare devices, the rest are operational.
- The downconverters select their input signals from a distributive switch matrix.
- The outputs of all downconverters are wired to the inputs of a distributive switching matrix.

In the event of a switchover, the input switch position of the primary device gets copied to the input switch of the backup device, then all downconverter parameters are mirrored and finally the output switch matrix is commanded to set all outputs which originally pointed to the primary downconverter now to the backup one.

The Downconverter-Protection-Controller device carries out the actual task of monitoring and redundancy switching. One Downconverter-Protection-Controller device is configured for each of the downconverter involved. Whether the individual downconverter is an operational or a spare device can be switched "on the fly" using a configuration parameter. In addition, a protection controller device can be configured to the INACTIVE state, which designates a device that is neither PRIMARY nor SPARE, e.g. as a placeholder for later extensions or if a device is temporarily unavailable.

Each Downconverter-Protection-Controller device monitors two configurable fault flags. The first fault is labeled "general fault" and should combine all fault which prevent a downconverter from being used. The second fault ("operational fault") should be active with fault conditions

that shall trigger a redundancy switching but shall not prevent a spare downconverter from being used if this fault is active. A missing input signal is an example for this type of fault.

When performing a redundancy switch, the device first requests a free spare device from the Protection-Switch-N-to-M device. It mirrors the settings of the device to be replaced on this and then sets the matrix outputs in such a way that the output of the spare downconverter is routed to the destinations that were previously served by the replaced downconverter.

The redundancy switch can also be triggered manually by clicking REPLACE-BY-SPARE. In the same way, after the replaced device has been repaired/checked, it can be switched back to operational mode (SWITCH-BACK). The original matrix configuration of the device is restored, but no downconverter parameters from the spare device are mirrored back to the primary one.

Controllers configured as SPARE may be bound to a certain primary downconverter. In this case the spare device is no longer part of the spare pool, instead it acts as a 1:1 backup for the selected primary downconverter. Use the 'reserved for' parameter to configure this feature. Controllers which have the 'reserved for' parameter set to NONE are part of the spare pool.

By default device parameters are mirrored after the backup downconverter input switch has been commanded and before the switch matrix is set to swap the downconverter output signals. For 1:1 downconverter pairs a different approach may be configured. If the '121MirrorMode' config parameter is set to IMMEDIATE at the primary controller, mirroring is skipped during the switchover procedure, thus speeding this up significantly. Instead mirroring the device parameters is triggered by the following events:

- A backup downconverter is configured as 'reserved for' for this primary downconverter.
- The operator clicks the MIRROR NOW button.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
protectionSwitch	The name of the Protection-Switch-N-To-M device, this Downconverter-Protection-Controller is assigned to.

parameter	*description*
downconverter	The name of the downconverter device this Downconverter-Protection-Controller monitors and controls.
role	The role of this device. One of PRIMARY, SPARE or INACTIVE.
matrixPort	The number of the matrix input the downconverter output is connected to. The number must be written with a leading zero for port numbers 01..09, the device expects the port number as it appears in the variable name of the matrix device driver.
faultId.01	The parameter ID to monitor for the "general fault". This fault is monitored by both, PRIMARY and SPARE devices. If a fault is pending, it triggers a redundancy switch or prevents a device being taken from the pool of spares, depending on the device's role.
faultId.02	The parameter ID to monitor for the "operational fault". This fault is recognized by PRIMARY devices, it also trigers an redundancy switch wif pending. The "operational fault" id ignored by SPARE devices.
preMirrorDelay	The time (msecs) between switching OFF all matrix routes originating from the SPARE mode before mirroring the parameters of the PRIMARY device to the SPARE one. This delay shall ensure that the SPARE device has a proper input signal when it gets configured for its next use.
postMirrorDelay	The time (msecs) between mirroring the parameters of the PRIMARY device to the SPARE one and mirroring the matrix port settings. This delay shall ensure that the SPARE device is completly configured before it gets conneceted to its destinations.
121MirrorMode	The way (SWITCHOVER, IMMEDIATE) the device handles the mirroring of modem parameters in a 1:1 redundancy configuration

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.protectionSwitch	TEXT	SAVE SETUP	StringRange
config.downconverter	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.role	CHOICE	SAVE SETUP	EnumRange (INACTIVE PRIMARY SPARE)
config.inputSwitch	TEXT	SAVE SETUP	StringRange
config.matrixPort	TEXT	SAVE SETUP	StringRange
config.faultId.01	TEXT	SAVE SETUP	StringRange
config.faultId.02	TEXT	SAVE SETUP	StringRange
config.preMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.postMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.121MirrorMode	CHOICE	SAVE SETUP	EnumRange (SWITCHOVER IMMEDIATE)
protection	CHOICE	SAVE	EnumRange (OFF ON)
replaceBySpare	TEXT	NOPRESET	StringRange
switchBack	TEXT	NOPRESET	StringRange
forceMirror	TEXT	NOPRESET	StringRange
reservedFor	CHOICE		EnumRange DIS (NONE)
comment	TEXT	SAVE	StringRange
reservedFor.save	TEXT	SAVE	StringRange
modemFault	BOOLEAN	R/O	BooleanRange R/O (true false)
externalFault	BOOLEAN	R/O	BooleanRange R/O (true false)
inUse	BOOLEAN	R/O	BooleanRange R/O (true false)
state	TEXT	R/O	StringRange R/O
mayForceMirror	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (D/C general)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (D/C operational)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.173 Dual-Timezone-Clock

The Dual-Timezone-Clock device displays the actual time in server time (UTC) and in in a second 'local' timezone which may be configured by the operator. The time offset between UTC and local time is configured at the device's maintenance page as a fixed number of hours, the device is not aware about daylight savings changes.

To make the device update its time once a second, the device should be configured as the only device in a device thread with the standard idle time of 1000 msec.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual time in UTC and in the configured 'local' timezone.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window you can define the time offset and the label for the 'local' time provided by the device.

parameter	*description*
offset	The offset to be added to UTC in order to result in the displayed local time. The offset either may be entered in hours (integer or floating point) or in a "HH:MM" format. Both formats may be preceded by a "-" sign.
description	This text is displayed in front of to the 'local' time field in the device screen of the device. You may enter a timezone name, the name of a city or any text which uniquely identifies the 'local' timezone.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.offset	TEXT	SAVE SETUP	StringRange
config.description	TEXT	SAVE SETUP	StringRange
utc.time	TEXT	R/O	StringRange R/O
utc.label	TEXT	R/O	StringRange R/O
local.time	TEXT	R/O	StringRange R/O
local.label	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Circuit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.174 Dual-Translator

The Dual-Translator logical device translates a set of user definable choices into the position settings of two switches (or equivalent). It may be used to combine two switches to a single function in the user interface.

To make use of the Dual-Translator, install it in the device setup, configure the translation tables and the message ID you want to translate. In your user screens operate the translated parameter now through the 'value' parameter of the Dual-Translator instead of directly via the parameters original message ID.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Simply contains a choice box to operate the translated parameter. Provided for completeness and for testing purposes.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Dual-Translator device with the device / parameter it shall translate.

parameter	*description*
choices	Enter the list of choices here which are to be presented to the operator when he edits the 'value' setting of this Dual-Translator device. The choice list must be entered as a comma separated list of tokens with no additional spaces between them.
messageId1	Enter the message identifier here of the first parameter / switch setting which shall be controlled, eg. "SWITCH-1. position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
choices1	Enter the list of choices/positions here for the first switch. There must be the same number of position settings contained in the list as choices defined above. Choices and switch positions are assigned by the order they appear in their lists.
messageId2	Enter the message identifier here of the second parameter / switch setting which shall be controlled, eg. "SWITCH-2. position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
choices2	Enter the list of choices/positions here for the second switch. There must be the same number of position settings contained in the list as choices defined above. Choices and switch positions are assigned by the order they appear in their lists.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.choices	TEXT	SAVE SETUP	StringRange
config.messageId1	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.choices1	TEXT	SAVE SETUP	StringRange
config.messageId2	TEXT	SAVE SETUP	StringRange
config.choices2	TEXT	SAVE SETUP	StringRange
value	CHOICE		EnumRange ()

9.175 Dual-Wind-Speed

The Dual-Wind-Speed logical device interprets two voltages measured by a PLC or a voltmeter as wind speed. It calculates average values from the measured speeds and compares all four measurement values (2 x momentary, 2 x average) against individual limits, raining a fault if one of the values is above its limit.

Beside this, the device treats the two measured wind speeds as redundant data sources. It expects the measured wind speeds to be almost the same and signals a fault if the measured values differ too much.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the measured wind speeds and their averages.
- --- Permits to edit fault limits.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Dual-Wind-Speed device.

parameter	*description*
input.1.id	Enter the parameter ID of the A/D converter providing the voltage for the first wind speed to measure.
input.1.scale	The scale value used to convert voltage to speed. The device expects the speed increase (km/h) that is equivalent to a 1 Volt voltage increase to be entered here.

parameter	*description*
input.1.offset	The offset value is is the wind speed (km/h) which corresponds to 0V measured voltage.
input.2.id	The parameter ID of the second voltage to measure.
input.2.scale	The second wind speed scale factor.
input.2.offset	The second wind speed offset.
averagingTime	The averaging time (1..30 minutes) used to calculate the averaged speed values. It applies to both average wind speeds.

Remarks

- The averaging filters are initialized to zero when the software starts. The average values then increase until they reach the momentary measurement values.
- All fault limits are set to 999.0 km/h to disable the limit checking by default.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input.1.id	TEXT	SAVE SETUP	StringRange
config.input.1.scale	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.input.1.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.input.2.id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input.2.scale	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.input.2.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.averagingTime	INTEGER	SAVE SETUP	IntegerRange (1 .. 30)
speed.1.actual	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed.1.avg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed.2.actual	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
speed.2.avg	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
limit.1.actual	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
limit.1.avg	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
limit.2.actual	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
limit.2.avg	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
limit.delta	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
limit.minSpeed	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Speed 1 actual limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Speed 1 avg limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Speed 2 actual limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Speed 2 avg limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Speed delta limit)

9.176 E2V-TWTA

This page describes the device driver and the device window for the E2V Stellar STA1000 series TWT amplifiers. which works also with the 1+1 redundancy indoor HPA controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF,Redundancy)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 48
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,7,E,1* .
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Hints

This drivers does not put the TWTA automatically into "remote mode" because there are some issues with the indoor controller. So to start controlling the TWTA from the M&C system you have to set the ctrlmode to remote by yourself.

Another point is, that you will get a communication fault if you send a command a second time again if it was already executed.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.totalElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.inTransmitElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.inStandbyElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.inOffElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY BEAM ON)
tx.gain	INTEGER		IntegerRange (0 .. 4095)
state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ctrlMode	CHOICE		EnumRange (REMOTE LOCAL)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0 .. 0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input	INTEGER	R/O	IntegerRange R/O (0 .. 0)
red.switch	CHOICE		EnumRange (ANTENNA LOAD)
red.mode	CHOICE		EnumRange (AUTOMATIC MANUAL)
internal.d1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.off	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.warmup	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.switch	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.mode	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High power)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HV overload)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Peak helix current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mean helix current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Reflected power)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PFC fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (10MHz reference)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Case temperature)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Main fan)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Stirrer fans)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Collector temp.)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Heater volts limit)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Collector volts limit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Cathode volts limit)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Internal comm)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.177 ECU-J0475

This page describes the device driver and the device window for the ECU Corporation J0475 Ruggedized Air Condition via SNMP. Beside setting the cooling setpoint and the en or disabling the unit it is readonly.

It was tested with SW revision: 4

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters and measurement readings.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple

lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.code	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
enable	CHOICE		EnumRange (OFF AUTO)
setpoint.cool	INTEGER		IntegerRange (0 .. 0)
setpoint.cool.f	INTEGER		IntegerRange (0 .. 0)
meas.pressure.dis	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.pressure.dis2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.pressure.suc	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.pressure.suc2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp.rack1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.rack2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.return	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.supply	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.temp.ambient	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.rack1.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.rack2.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.return.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.supply.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.ambient.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state	CHOICE	R/O	EnumRange R/O (OFF FAN COOL HEAT)
state.fan	BOOLEAN	R/O	BooleanRange R/O (true false)
state.cool	BOOLEAN	R/O	BooleanRange R/O (true false)
state.heat	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.setpoint	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary/Global)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Rack 1 Temp Sens)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Rack 2 Temp Sens)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Shelter Humidity Sens)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Compressor VFD)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Condenser Fan)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Evaporator Fan)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Discharge Pressure Transducer Sens)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Return Air Temp Sens)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Outside Air Temperature Sensor)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Supply Air Temperature Sens)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Hot Room)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cold Room)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High Temperature Cutout Switch)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Air Flow Evaporator blower)
faults.16	ALARM	R/O	AlarmFlagRange R/O (High Pressure Transducer)
faults.17	ALARM	R/O	AlarmFlagRange R/O (High Pressure Switch)
faults.18	ALARM	R/O	AlarmFlagRange R/O (High Pressure DP Transducer)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Low Pressure Transducer)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Low Pressure Switch)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Low Pressure Suct PT Transducer)

9.178 EF-Data-SDM300

This page describes the device driver and the device window for the EF-Data SDM Satellite Modem. The driver actually supports the following modem operation modes:

- SCPC
- IDR overhead
- IBS overhead
- ASYNC overhead

The Drop/Insert mode is not yet supported by the driver, nor are the EF-Data AUPC and MUX options.

Not all SDM300 modems support all functions which are implemented in the device driver. If you try to select a function which is unsupported by this particular modem, the parameter will fall back to it's old value. Each missing attempt to set a parameter is recorded to the event log.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The Reed-Solomon settings.
- --- The IDR/IBS mode settings.
- --- The ASYNC overhead settings.
- --- The data interface settings
- --- The receive quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the SDM300.

parameter	***description***
address	The modem's RS485 packet address. Valid addresses are 1..255, 1 is the factory default. You must set the same address than set at the modem's front panel
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,E,7,1* . You have to set the same value at the device's front panel.

Remote / local switching

The SDM300 does not provide a local/remote parameter which can be set at the device. Instead, the SDM300 is set to remote control with a command sent by the M&C/VLC. Touching one of the buttons at the front panel normally reverts the modem to local front panel control. The software repeatedly sends the "REMOTE" command to be sure that the SDM300 is in remote mode at any time. Hence, if you want to operate a SDM300 which is connected to a sat-nms system at it's front panel, pull out the remote interface plug from or set the device 'out of service' at the software's user interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.id	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.card.1	TEXT	R/O	StringRange R/O
info.card.2	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
modemType	CHOICE		EnumRange (IDR IBS DI ASYNC EFD CUSTOM)
duplexMode	CHOICE		EnumRange (TX_ONLY RX_ONLY DUPLEX)
refClkSrc	CHOICE		EnumRange (INT EXT1 EXT5 EXT10 EXT20)
refClkFreq	FLOAT		DoubleRange (8.0 .. 10000.0)
loopBack	CHOICE		EnumRange (OFF IFC BASEB IF RF)
tx.frequency	FLOAT		DoubleRange (55.000000 .. 188.000000)
tx.power	FLOAT		DoubleRange (-30.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
tx.mod.symbolRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.mod.type	CHOICE		EnumRange (SEQ VIT)
tx.mod.fec	CHOICE		EnumRange (QPSK-1/2 QPSK-3/4 QPSK-7/8 BPSK-1/2)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT REF)
tx.mod.cwMode	CHOICE		EnumRange (OFF DUAL OFFSET CENTER)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.scrambling	CHOICE		EnumRange (OFF ON)
tx.mod.filter	CHOICE		EnumRange (INTL EFD CSC FDC SDM51)
tx.mod.differential	CHOICE		EnumRange (OFF ON)
tx.rs.bypass	CHOICE		EnumRange (OFF ON)
tx.rs.interleave	CHOICE		EnumRange (4 8 16)
tx.ifc.clockPhase	CHOICE		EnumRange (NRM INV AUTO)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.overheadType	CHOICE		EnumRange (NONE IDR IBS DI ASYNC)
tx.ifc.framingMode	CHOICE		EnumRange (AMI HDB3 B8ZS)
tx.ifc.hardware	CHOICE		EnumRange (G703 V35 RS422 RS232)
tx.asy.baudrate	CHOICE		EnumRange (110 150 300 600 1200 2400 4800 9600 19200 38400)
tx.asy.dataBits	CHOICE		EnumRange (5 6 7 8)
tx.asy.stopBits	CHOICE		EnumRange (1 2)
tx.asy.parity	CHOICE		EnumRange (ODD EVEN NONE)
tx.asy.hardware	CHOICE		EnumRange (RS232 RS485 RS485_2W)
tx.idr.overheadType	CHOICE		EnumRange (DATA AUDIO)

name	*type*	*flags*	*range*
tx.idr.bwAlarm.1	CHOICE		EnumRange (OFF ON)
tx.idr.bwAlarm.2	CHOICE		EnumRange (OFF ON)
tx.idr.bwAlarm.3	CHOICE		EnumRange (OFF ON)
tx.idr.bwAlarm.4	CHOICE		EnumRange (OFF ON)
tx.idr.audioLevel.1	INTEGER		IntegerRange (-20 .. 10)
tx.idr.audioLevel.2	INTEGER		IntegerRange (-20 .. 10)
rx.frequency	FLOAT		DoubleRange (50.000000 .. 180.000000)
rx.dem.dataRate	INTEGER		IntegerRange (0 .. 0)
rx.dem.symbolRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.dem.type	CHOICE		EnumRange (SEQ VIT)
rx.dem.fec	CHOICE		EnumRange (QPSK-1/2 QPSK-3/4 QPSK-7/8 BPSK-1/2)
rx.dem.clockSource	CHOICE		EnumRange (INT EXT SAT REF INS)
rx.dem.spectrumInvert	CHOICE		EnumRange (NRM INV)
rx.dem.scrambling	CHOICE		EnumRange (OFF ON)
rx.dem.filter	CHOICE		EnumRange (INTL EFD CSC FDC)
rx.dem.differential	CHOICE		EnumRange (OFF ON)
rx.dem.sweepMode	INTEGER		IntegerRange (0 .. 999)
rx.dem.sweepOffset	INTEGER		IntegerRange (-35000 .. 35000)
rx.dem.sweepRange	INTEGER		IntegerRange (0 .. 70000)
rx.buf.size	INTEGER		IntegerRange (0 .. 99)
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.ber	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.bufFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rx.limit.ber	CHOICE		EnumRange (NONE 1E-3 1E-4 1E-5 1E-6 1E-7 1E-8)
rx.rs.bypass	CHOICE		EnumRange (OFF ON CORR)
rx.rs.interleave	CHOICE		EnumRange (4 8 16)
rx.ifc.clockPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
rx.ifc.overheadType	CHOICE		EnumRange (NONE IDR IBS DI ASYNC)
rx.ifc.framingMode	CHOICE		EnumRange (AMI HDB3 B8ZS)
rx.ifc.hardware	CHOICE		EnumRange (G703 V35 RS422 RS232)
rx.asy.baudrate	CHOICE		EnumRange (110 150 300 600 1200 2400 4800 9600 19200 38400)
rx.asy.dataBits	CHOICE		EnumRange (5 6 7 8)
rx.asy.stopBits	CHOICE		EnumRange (1 2)
rx.asy.parity	CHOICE		EnumRange (ODD EVEN NONE)
rx.asy.hardware	CHOICE		EnumRange (RS232 RS485)
rx.idr.overheadType	CHOICE		EnumRange (DATA AUDIO)
rx.idr.bwAlarm.1	CHOICE		EnumRange (OFF ON)
rx.idr.bwAlarm.2	CHOICE		EnumRange (OFF ON)
rx.idr.bwAlarm.3	CHOICE		EnumRange (OFF ON)
rx.idr.bwAlarm.4	CHOICE		EnumRange (OFF ON)
rx.idr.audioLevel.1	INTEGER		IntegerRange (-20 .. 10)
rx.idr.audioLevel.2	INTEGER		IntegerRange (-20 .. 10)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Modulator Module)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Modulator IF synth)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Data clock synth)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Mod. I-channel)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Mod. Q-channel)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Mod. AGC level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reference PLL lock)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Reference activity)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Demodulator Module)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Decoder lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Demod. IF synth)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Demod. I-channel)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Demod. Q-channel)
faults.14	ALARM	R/O	AlarmFlagRange R/O (BER threshold)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tx-ifc data/AIS)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Tx-ifc PLL lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Tx-ifc clk activity)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Drop)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Tx audio 1 clip)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Tx audio 2 clip)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Insert)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Buffer underflow)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Buffer overflow)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Rx-ifc data/AIS)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Frame BER)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Rx backward alarm)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Buffer clk activity)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Buffer clk PLL lock)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Demux lock)
faults.30	ALARM	R/O	AlarmFlagRange R/O (2047 pat. lock detect)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Buffer full)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Rx audio 1 clip)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Rx audio 2 clip)
faults.34	ALARM	R/O	AlarmFlagRange R/O (M&C Module)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Battery)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Power supply -12V)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #1)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #2)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #3)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Tx Alarm #4)

name	*type*	*flags*	*range*
faults.44	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #1)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #2)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #3)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Rx Alarm #4)
internal.cs	TEXT	R/O	StringRange R/O
internal.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.c	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.g	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.j	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.k	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.l	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.m	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.n	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.o	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.q	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.179 ET-ADH-2A

This page describes the device driver and the device window for the Environmental Technology ADH-2A waveguide pressurizer / dehydrator. The device driver for this device operates read

only, there are no setting which can be made.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays meter readings like device state, pressure and temperature.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	The device address (decimal). The factory default is 48
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the dehydrator is *9600,7,E,1* .

Remarks

The ADH-2A uses a RS422 serial interface, you will need a RS232 to RS422 converter to control the device from a standard RS232 port.

There are a couple of jumpers / switches inside the device (please refer to the ADH-2A operation manual). The driver expects the following settings:

- Device address 0x30 (48 decimal)
- baud rate 9600 (you may use 1200 as well if you configure the VLC/M&C to the same baud rate, but this will slow down the communication significantly.)
- Metric display. If the device is configured to english display, readings will be in*°F* , *psi* etc. but the sat-nms software will still append metric units to the reading.

The ADH-2A operation manual also describes a redundant configuration of two ADH-2A units, connected through their RS422 ports. The sat-nms software is not capable to monitor the ADH-2A in this configuration.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
internal.s	TEXT	R/O	StringRange R/O
internal.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
pressure2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature	FLOAT	R/O	DoubleRange R/O (0 .. 0)
temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
absorptionUnit1	CHOICE	R/O	EnumRange R/O (IDLE ACTIVE REGENERATING FULL DEAD LEAKY STANDBY)
absorptionUnit2	CHOICE	R/O	EnumRange R/O (IDLE ACTIVE REGENERATING FULL DEAD LEAKY STANDBY)
dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flowRate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (System leaky)
faults.04	ALARM	R/O	AlarmFlagRange R/O (High temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low line voltage)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (AU#1 heat)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AU#1 cool)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AU#2 heat)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AU#2 cool)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Leaky shutdown)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Dew point)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Change compressor)

9.180 ETI-ADH-NETCOM

This page describes the device driver and the device window for the Environmental Technology Inc. wave guide pressurizer / dehydrator ADH-NETCOM via SNMP

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.runTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.unitTemp	TEXT	R/O	StringRange R/O
internal.unitPres	TEXT	R/O	StringRange R/O
internal.unit	TEXT	R/O	StringRange R/O
internal.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pressure	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
absorptionUnit1	CHOICE	R/O	EnumRange R/O (IDLE IN USE REGENERATING)
absorptionCond1	CHOICE	R/O	EnumRange R/O (OK FULL DEAD UNKNOWN)
absorptionUnit2	CHOICE	R/O	EnumRange R/O (IDLE IN USE REGENERATING)
absorptionCond2	CHOICE	R/O	EnumRange R/O (OK FULL DEAD UNKNOWN)
dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dutyCycleTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low Pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High duty cycle)
faults.04	ALARM	R/O	AlarmFlagRange R/O (High temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Canister 1 no heat)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Canister 1 no cool)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Canister 2 no heat)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Canister 2 no cool)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unable to pressurize)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Dew point alarm)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Canister 1 thermistor)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Canister 2 thermistor)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Ambient thermistor)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Overpressure)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Bad cal. in EEPROM)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Bad limits in EEPROM)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Short cycling)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Pump shutdown)

9.181 ETL-2773-XXYY

This page describes the SNMP based readonly device driver for the 2773-XXYY Series LNB Tone switching and DC power supply from ETL Systems.

The driver was tested with the following model:

- with 16 modules
- SW release e148 1.0

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The I/O controller parameters and sensor measurements
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	description
address	The SNMP address specification for the device (details below).
moduleName 01...16	Defines the module alias name stored only locally on the M&C.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` and `private` as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.module.01.name	TEXT	SAVE SETUP	StringRange
config.module.02.name	TEXT	SAVE SETUP	StringRange
config.module.03.name	TEXT	SAVE SETUP	StringRange
config.module.04.name	TEXT	SAVE SETUP	StringRange
config.module.05.name	TEXT	SAVE SETUP	StringRange
config.module.06.name	TEXT	SAVE SETUP	StringRange
config.module.07.name	TEXT	SAVE SETUP	StringRange
config.module.08.name	TEXT	SAVE SETUP	StringRange
config.module.09.name	TEXT	SAVE SETUP	StringRange
config.module.10.name	TEXT	SAVE SETUP	StringRange
config.module.11.name	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.module.12.name	TEXT	SAVE SETUP	StringRange
config.module.13.name	TEXT	SAVE SETUP	StringRange
config.module.14.name	TEXT	SAVE SETUP	StringRange
config.module.15.name	TEXT	SAVE SETUP	StringRange
config.module.16.name	TEXT	SAVE SETUP	StringRange
config.deviceComment	TEXT	SAVE SETUP	StringRange
module.01.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.02.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.03.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.04.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.05.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.06.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.07.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.08.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.09.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.10.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.11.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.12.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.13.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.14.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.15.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.16.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
module.01.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.02.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.03.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.04.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.05.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.06.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.07.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.08.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.09.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.10.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.11.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.12.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.13.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.14.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.15.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.16.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
setMonitoring	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Module 1 Low Current)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Module 1 High Current)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 2 Low Current)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 2 High Current)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Module 3 Low Current)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Module 3 High Current)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Module 4 Low Current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Module 4 High Current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module 5 Low Current)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module 5 High Current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Module 6 Low Current)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Module 6 High Current)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Module 7 Low Current)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Module 7 High Current)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Module 8 Low Current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Module 8 High Current)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Module 9 Low Current)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Module 9 High Current)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Module 10 Low Current)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Module 10 High Current)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Module 11 Low Current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Module 11 High Current)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Module 12 Low Current)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Module 12 High Current)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Module 13 Low Current)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Module 13 High Current)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Module 14 Low Current)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Module 14 High Current)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Module 15 Low Current)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Module 15 High Current)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Module 16 Low Current)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Module 16 High Current)

9.182 ETL-Alto-Amp

This page describes the device driver and the device window for ETL Alto LNA switching unit.

Please note: Monitoring of more than one amplifier per module is not implemented.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch parameters.
- --- Amplifier readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- standbyMode.1 --- Defines the standby mode for the 1st module.
- standbyMode.2 --- Defines the standby mode for the 2nd module.
- standbyMode.3 --- Defines the standby mode for the 3rd module (standby module).

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.standbyMode.1	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyMode.2	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyMode.3	CHOICE	SETUP	EnumRange (COLD HOT)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)
priority	CHOICE		EnumRange (NONE PATH-1 PATH-2)
protection	CHOICE		EnumRange (MANUAL AUTO)
protection.1	CHOICE		EnumRange (MANUAL AUTO)
protection.2	CHOICE		EnumRange (MANUAL AUTO)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cpuTemp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cpuTemp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cpuTemp.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.15v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.15v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.15v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.12v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.12v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.12v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.swVersion	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.module.1	TEXT	R/O	StringRange R/O
info.module.1.version	TEXT	R/O	StringRange R/O
info.module.2	TEXT	R/O	StringRange R/O
info.module.2.version	TEXT	R/O	StringRange R/O
info.module.3	TEXT	R/O	StringRange R/O
info.module.3.version	TEXT	R/O	StringRange R/O
psu.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.vbus	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.6v	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.5v	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.fan	FLOAT	R/O	DoubleRange R/O (0 .. 0)
psu.2.fan	FLOAT	R/O	DoubleRange R/O (0 .. 0)
chassis.fan.1	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
chassis.fan.2	FLOAT	R/O	DoubleRange R/O (0 .. 0)
chassis.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Chassis Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch position)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Humidity)
faults.09	ALARM	R/O	AlarmFlagRange R/O (VBus Voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (6.5Volt)
faults.11	ALARM	R/O	AlarmFlagRange R/O (3Volt)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Module 1 Communication)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Module 2 Communication)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Module 3 Communication)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Module 1 Switch status)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Module 2 Switch status)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Module 3 Switch status)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Module 1 Fuse)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Module 2 Fuse)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (Module 3 Fuse)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Module 1 CPU Temperature)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Module 2 CPU Temperature)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Module 3 CPU Temperature)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Module 1 Status)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Module 2 Status)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Module 3 Status)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Module 1 Current)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Module 2 Current)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Module 3 Current)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Module 1 15Volt)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Module 2 15Volt)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Module 3 15Volt)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Module 1 12Volt)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Module 2 12Volt)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Module 3 12Volt)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Module 1 6.5Volt)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Module 2 6.5Volt)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Module 3 6.5Volt)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Module 1 5Volt)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Module 2 5Volt)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Module 3 5Volt)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Module 1 Temperature)

name	*type*	*flags*	*range*
faults.45	ALARM	R/O	AlarmFlagRange R/O (Module 2 Temperature)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Module 3 Temperature)
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O

9.183 ETL-Alto-DA

This page describes the *ETL-Alto-DA* device driver and its device window. The*ETL-Alto-DA* device driver monitors and controls the following devices:

ETL Systems Alto L-band 1+1 Redundant Amplifier Chassis with monitor ports, 50 ohm

- Model: **ALT-C319-1U**

ETL Systems L-band Combiner Active 16 Way with Hot Swap, Front Mounted Dual Redundant Amplifiers

- Model: **C1601S1ULA-22532**

ETL Systems L-band Splitter active 16 way with variable gain, variable slope, hot swap dual redundant amplifiers & hot-swap power supplies

- Model: **D0116S1ULA-22522**

The driver was tested with the following versions:

- 2 x ALT-R-L1-012 module
- Module SW version e3471v6
- CPU SW version: e225 2v1
- Chassis SW version: e567 1v9
- Chassis SW version: e567 1v9

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules gain/slope and redundancy settings.
- --- The modules,amplifier and chassis voltage and temperature readings.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- standbyMode.1 --- Defines the hot standby mode for module 1. If selected the redundant path will always be powered.
- standbyMode.2 --- Defines the hot standby mode for module 2. If selected the redundant path will always be powered.
- standbyMode.3 --- Defines the hot standby mode for module 3. If selected the redundant path will always be powered.
- standbyMode.4 --- Defines the hot standby mode for module 4. If selected the redundant path will always be powered.
- standbyCheck.1 --- Defines the check interval in cold standby mode for module 1.
- standbyCheck.2 --- Defines the check interval in cold standby mode for module 2.
- standbyCheck.3 --- Defines the check interval in cold standby mode for module 3.
- standbyCheck.4 --- Defines the check interval in cold standby mode for module 4.

- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.chassis.version	TEXT	R/O	StringRange R/O
info.modules	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.module.1.model	TEXT	R/O	StringRange R/O
info.module.1.version	TEXT	R/O	StringRange R/O
info.module.2.model	TEXT	R/O	StringRange R/O
info.module.2.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.standbyMode.1	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyMode.2	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyMode.3	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyMode.4	CHOICE	SETUP	EnumRange (COLD HOT)
config.standbyCheck.1	CHOICE	SETUP	EnumRange (MANUAL HOURLY DAILY WEEKLY)

name	*type*	*flags*	*range*
config.standbyCheck.2	CHOICE	SETUP	EnumRange (MANUAL HOURLY DAILY WEEKLY)
config.standbyCheck.3	CHOICE	SETUP	EnumRange (MANUAL HOURLY DAILY WEEKLY)
config.standbyCheck.4	CHOICE	SETUP	EnumRange (MANUAL HOURLY DAILY WEEKLY)
config.deviceComment	TEXT	SAVE SETUP	StringRange
position.1	CHOICE		EnumRange (A B)
location.1	CHOICE	R/O	EnumRange R/O (PATH-A PATH-B)
protection.1	CHOICE		EnumRange (MANUAL AUTO)
path.gain.1	INTEGER		IntegerRange (-3 .. 27)
path.slope.1	INTEGER		IntegerRange (0 .. 8)
alias.1	TEXT	SAVE	StringRange
position.2	CHOICE		EnumRange (A B)
location.2	CHOICE	R/O	EnumRange R/O (1B 2B)
protection.2	CHOICE		EnumRange (MANUAL AUTO)
path.gain.2	INTEGER		IntegerRange (-3 .. 27)
path.slope.2	INTEGER		IntegerRange (0 .. 8)
alias.2	TEXT	SAVE	StringRange
gain.1	INTEGER		IntegerRange (-3 .. 27)
gain.2	INTEGER		IntegerRange (-3 .. 27)
gain.3	INTEGER		IntegerRange (-3 .. 27)
gain.4	INTEGER		IntegerRange (-3 .. 27)
slope.1	INTEGER		IntegerRange (0 .. 8)
slope.2	INTEGER		IntegerRange (0 .. 8)
slope.3	INTEGER		IntegerRange (0 .. 8)
slope.4	INTEGER		IntegerRange (0 .. 8)

name	*type*	*flags*	*range*
meas.voltage.1.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.1.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.1.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.1.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.1.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.1.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.2.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.2.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.3.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.3.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.4.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.4.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.temp.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.6v.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.5v.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.neg2v.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.neg2v.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.neg2v.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.neg2v.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
monitoring.1	TEXT	NOPRESET	StringRange
monitoring.2	TEXT	NOPRESET	StringRange
monitoring.3	TEXT	NOPRESET	StringRange
monitoring.4	TEXT	NOPRESET	StringRange
psu.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.vbus	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.6v	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.5v	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chassis.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.psu.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.psu.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Chassis Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBus Voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (6.5Volt)
faults.06	ALARM	R/O	AlarmFlagRange R/O (5Volt)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch position)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Module 1 Communication)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Module 2 Communication)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Module 3 Communication)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Module 4 Communication)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amp 1)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amp 1)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amp 1)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amp 1)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amp 2)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amp 2)

name	*type*	*flags*	*range*
faults.30	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amp 2)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amp 2)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amp 3)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amp 3)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amp 3)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amp 3)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amp 4)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amp 4)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amp 4)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amp 4)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amp 5)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amp 5)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amp 5)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amp 5)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Module 1 6.5Volt)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Module 2 6.5Volt)

name	*type*	*flags*	*range*
faults.50	ALARM	R/O	AlarmFlagRange R/O (Module 3 6.5Volt)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Module 4 6.5Volt)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Module 1 5Volt)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Module 2 5Volt)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Module 3 5Volt)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Module 4 5Volt)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Module 1 -2Volt)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Module 2 -2Volt)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Module 3 -2Volt)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Module 4 -2Volt)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Module 1 Temperature)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Module 2 Temperature)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Module 3 Temperature)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Module 4 Temperature)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Module 1 Switch status)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Module 2 Switch status)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Module 3 Switch status)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Module 4 Switch status)

name	*type*	*flags*	*range*
faults.80	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.81	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Protection 2)
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O

9.184 ETL-Dextra-Combiner

This page describes the device driver and the device window for ETL Dextra Dual 4-way combiner/Dual 8-way combiner.

It was tested with following model and release.

- Model 22421 SW: e51 v1.8

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- In/Outputs.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- default.path.1 --- Defines the default auto mode main path of combiner 1 .
- default.path.2 --- Defines the default auto mode main path of combiner 2 .
- hotStandbyMode.1 --- Defines the hot standby mode of combiner 1 If selected the redundant path will always be powered
- hotStandbyMode.2 --- Defines the hot standby mode of combiner 2. If selected the redundant path will always be powered
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.unitTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.psuAVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuBVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vbusVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.3.3Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.default.path.1	CHOICE	SETUP	EnumRange (PATH-A PATH-B N/A)
config.default.path.2	CHOICE	SETUP	EnumRange (PATH-A PATH-B N/A)
config.hotStandbyMode.1	CHOICE	SETUP	EnumRange (DISABLED ENABLED N/A)
config.hotStandbyMode.2	CHOICE	SETUP	EnumRange (DISABLED ENABLED N/A)
config.deviceComment	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
protection.1	CHOICE		EnumRange (AUTO MANUAL)
protection.2	CHOICE		EnumRange (AUTO MANUAL)
location.1	CHOICE	R/O	EnumRange R/O (PATH-A PATH-B)
position.1	CHOICE		EnumRange (A B)
location.2	CHOICE	R/O	EnumRange R/O (PATH-A PATH-B)
position.2	CHOICE		EnumRange (A B)
unit.01.name	TEXT		StringRange
unit.02.name	TEXT		StringRange
amp.01.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.02.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.03.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.04.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
amp.01.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
amp.02.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
amp.03.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
amp.04.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.04	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (3.3V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Combiner 1 Path A)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Combiner 1 Path B)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Combiner 2 Path A)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Combiner 2 Path B)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Protection 1)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Protection 2)

9.185 ETL-Dextra-DistAmp

This page describes the device driver and the device window for ETL Dextra 10MHz Distribution Amplifier Type 24282. Monitoring of more than one amplifier per module is not implemented.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- In/Outputs.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- gainMode --- Defines the outputs gain mode. Automatic, Low or High.
- outputs --- Defines the number of outputs.
- defaultInput --- Defines the default input path.
- defaultPath --- Defines the default amplifier path.
- hotStandbyMode --- Defines the hot standby mode. If selected the redundant path will always be powered.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.unitTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuAVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuBVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.vbusVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.12Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5.4Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.3.3Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.gainMode	CHOICE	SETUP	EnumRange (AUTO LOW HIGH)
config.outputs	CHOICE	SAVE SETUP	EnumRange (16 32)
config.default.input	CHOICE	SETUP	EnumRange (INPUT 1 INPUT 2)
config.default.path	CHOICE	SETUP	EnumRange (PATH A PATH B)
config.hotStandbyMode	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
internal.pathAIndex	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.pathBIndex	INTEGER	R/O	IntegerRange R/O (0 .. 0)
protection	CHOICE		EnumRange (AUTO MANUAL INPUT 1 MANUAL INPUT 2)
location	CHOICE	R/O	EnumRange R/O (INPUT 1 INPUT 2)
position	CHOICE	R/O	EnumRange R/O (A B)
switch.state	CHOICE	R/O	EnumRange R/O (OK INPUT 1 FAILED INPUT 2 FAILED INPUT 1 & 2 FAILED)
protection.1	CHOICE		EnumRange (AUTO MANUAL PATH A MANUAL PATH B)
location.1	CHOICE	R/O	EnumRange R/O (PATH A PATH B)
position.1	CHOICE	R/O	EnumRange R/O (A B)
switch.1.state	CHOICE	R/O	EnumRange R/O (OK PATH A FAILED PATH B FAILED PATH A & B FAILED)
internal.loc	TEXT	R/O	StringRange R/O
internal.in.stat	TEXT	R/O	StringRange R/O
gain	CHOICE	R/O	EnumRange R/O (LOW HIGH)
input.01.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
input.02.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
input.01.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
input.02.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.01.attn	INTEGER		IntegerRange (0 .. 10)
output.02.attn	INTEGER		IntegerRange (0 .. 10)
output.03.attn	INTEGER		IntegerRange (0 .. 10)
output.04.attn	INTEGER		IntegerRange (0 .. 10)
output.05.attn	INTEGER		IntegerRange (0 .. 10)
output.06.attn	INTEGER		IntegerRange (0 .. 10)
output.07.attn	INTEGER		IntegerRange (0 .. 10)
output.08.attn	INTEGER		IntegerRange (0 .. 10)
output.09.attn	INTEGER		IntegerRange (0 .. 10)
output.10.attn	INTEGER		IntegerRange (0 .. 10)
output.11.attn	INTEGER		IntegerRange (0 .. 10)
output.12.attn	INTEGER		IntegerRange (0 .. 10)
output.13.attn	INTEGER		IntegerRange (0 .. 10)
output.14.attn	INTEGER		IntegerRange (0 .. 10)
output.15.attn	INTEGER		IntegerRange (0 .. 10)
output.16.attn	INTEGER		IntegerRange (0 .. 10)
output.01.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.02.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)

name	*type*	*flags*	*range*
output.03.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.04.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.05.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.06.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.07.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.08.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.09.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.10.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.11.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.12.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.13.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.14.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.15.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.16.level	FLOAT	R/O	DoubleRange R/O (-17.0 .. 25.0)
output.01.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.02.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.03.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.04.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.05.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.06.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.07.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.08.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.09.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.10.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.11.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.12.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.13.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.14.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.15.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.16.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
path.a.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
path.b.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.01.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.02.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.03.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.04.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.05.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.06.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.07.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.08.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.09.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.10.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.11.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.12.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.13.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.14.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.15.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.16.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
input.01.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.01.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.02.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.02.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)

name	*type*	*flags*	*range*
output.01.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.01.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.02.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.02.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.03.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.03.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.04.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.04.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.05.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.05.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.06.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.06.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.07.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.07.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.08.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.08.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.09.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.09.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.10.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.10.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.11.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.11.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.12.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.12.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.13.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.13.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)

name	*type*	*flags*	*range*
output.14.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.14.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.15.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.15.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.16.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.16.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.04	ALARM	R/O	AlarmFlagRange R/O (12V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5.4V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (3.3V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PATH A STATUS)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PATH B STATUS)
faults.10	ALARM	R/O	AlarmFlagRange R/O (INPUT 1 STATUS)
faults.11	ALARM	R/O	AlarmFlagRange R/O (INPUT 2 STATUS)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PORT 1 AMP STATUS)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PORT 2 AMP STATUS)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PORT 3 AMP STATUS)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PORT 4 AMP STATUS)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PORT 5 AMP STATUS)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PORT 6 AMP STATUS)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (PORT 7 AMP STATUS)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PORT 8 AMP STATUS)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PORT 9 AMP STATUS)
faults.21	ALARM	R/O	AlarmFlagRange R/O (PORT 10 AMP STATUS)
faults.22	ALARM	R/O	AlarmFlagRange R/O (PORT 11 AMP STATUS)
faults.23	ALARM	R/O	AlarmFlagRange R/O (PORT 12 AMP STATUS)
faults.24	ALARM	R/O	AlarmFlagRange R/O (PORT 13 AMP STATUS)
faults.25	ALARM	R/O	AlarmFlagRange R/O (PORT 14 AMP STATUS)
faults.26	ALARM	R/O	AlarmFlagRange R/O (PORT 15 AMP STATUS)
faults.27	ALARM	R/O	AlarmFlagRange R/O (PORT 16 AMP STATUS)
faults.44	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 1 STATUS)
faults.45	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 2 STATUS)
faults.46	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 3 STATUS)
faults.47	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 4 STATUS)
faults.48	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 5 STATUS)
faults.49	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 6 STATUS)
faults.50	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 7 STATUS)
faults.51	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 8 STATUS)
faults.52	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 9 STATUS)
faults.53	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 10 STATUS)

name	*type*	*flags*	*range*
faults.54	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 11 STATUS)
faults.55	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 12 STATUS)
faults.56	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 13 STATUS)
faults.57	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 14 STATUS)
faults.58	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 15 STATUS)
faults.59	ALARM	R/O	AlarmFlagRange R/O (OUTPUT 16 STATUS)

9.186 ETL-Genus-100MHz

This page describes the *ETL-Genus-100MHz* device driver and its device window. The *ETL-Genus-100MHz* device driver monitors and controls a 10MHz reference module installed in a ETL Genus chassis.

The driver was tested with the following equipment:

- Genus Frame 196-2
  - SW version: e573 1v22

and card models:

- OSC0023
  - SW version: e573 1v22

The different available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module
- ETL-Genus-10MHz monitors and controls one OSC 10MHz reference module
- ETL-Genus-Upconverter monitors and controls one Falcon series upconverter module
- ETL-Genus-Downconverter monitors and controls one Falcon series downconverter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
setLimits	En- or Disables Limit Alarm Monitoring
cpuReferenceSet	En or Disable the CPU reference set
internalSource	En- or Disable the internal reference source
reference	Defines is internal or external reference should be used
reference.loLimit	Defines the lower limit
reference.hiLimit	Defines the upper limit
slot	Defines the table index the 10MHz modules is using within the ETL module table. Default value is 35
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.vBus	FLOAT		DoubleRange (0.0 .. 0.0)
info.5v	FLOAT		DoubleRange (0.0 .. 0.0)
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.reference.enable	CHOICE	SETUP	EnumRange (OFF ON)
config.reference.dist	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.reference.loLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.reference.hiLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.slot	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
port.01.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.02.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.03.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.04.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.05.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.06.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.07.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.08.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.09.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.10.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.11.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.12.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.13.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.14.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.15.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
port.16.enable	CHOICE	R/O	EnumRange R/O (ON OFF)
protection	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
reference	CHOICE		EnumRange (INTERNAL EXT 10MHz AUTO EXT 100MHz)
meas.rfLevel.ext	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.refFreq	TEXT	R/O	StringRange R/O
internal.refSource	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PLL Lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBus)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (External RF)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.187 ETL-Genus-10MHz

This page describes the *ETL-Genus-10MHz* device driver and its device window. The *ETL-Genus-10MHz* device driver monitors and controls a 10MHz reference module installed in a ETL Genus chassis.

The driver was tested with the following equipment:

- Genus Frame 196-2
  - SW version: e573 1v22

and card models:

- OSC0023
  - SW version: e573 1v22

The different available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401

- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module
- ETL-Genus-100MHz monitors and controls one OSC 100MHz reference module
- ETL-Genus-Upconverter monitors and controls one Falcon series upconverter module
- ETL-Genus-Downconverter monitors and controls one Falcon series downconverter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
setLimits	En- or Disables Limit Alarm Monitoring
cpuReferenceSet	En or Disable the CPU reference set
internalSource	En- or Disable the internal reference source
reference	Defines is internal or external reference should be used
reference.loLimit	Defines the lower limit
reference.hiLimit	Defines the upper limit
slot	Defines the table index the 10MHz modules is using within the ETL module table. Default value is 35

parameter	description
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.vBus	FLOAT		DoubleRange (0.0 .. 0.0)
info.5v	FLOAT		DoubleRange (0.0 .. 0.0)
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.cpuReferenceSet	CHOICE	SETUP	EnumRange (ON OFF)
config.internalSource	CHOICE	SETUP	EnumRange (OFF ON)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.reference.loLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.reference.hiLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.slot	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
port.01.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.02.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.03.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.04.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.05.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.06.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.07.enable	CHOICE	SETUP	EnumRange (ON OFF)

name	*type*	*flags*	*range*
port.08.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.09.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.10.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.11.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.12.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.13.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.14.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.15.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.16.enable	CHOICE	SETUP	EnumRange (ON OFF)
port.01.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.02.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.03.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.04.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.05.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.06.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.07.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.08.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.09.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.10.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.11.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.12.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.13.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.14.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.15.level	FLOAT		DoubleRange (0.00 .. 0.00)
port.16.level	FLOAT		DoubleRange (0.00 .. 0.00)
protection	CHOICE		EnumRange (AUTO MANUAL)
meas.current.com	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meas.current.cpu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.rfLevel.ext	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.rfLevel.int	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (VBus)
faults.04	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (External RF)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal RF)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amp Common)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amp CPU)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amp 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amp 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amp 3)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amp 4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amp 5)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Amp 6)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Amp 7)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Amp 8)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Amp 9)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Amp 10)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Amp 11)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Amp 12)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Amp 13)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Amp 14)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Amp 15)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Amp 16)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.188 ETL-Genus-ALT

This page describes the *ETL-Genus-ALT* device driver and its device window. The *ETL-Genus-ALT* device driver monitors and controls one ALTO series amplifier card within a Genus Chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:

- Genus Frame
  - SW version: e530 2v14

and card models:

- ALT-G1S-S3-100A
  - SW version: e578 1v1

The different available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).

parameter	description
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35
monitoring	En- or Disables Alarm Monitoring
hasLinearityMode	En or Disable the linearity mode settings
hasLowGainMode	En- or Disable the low gain mode settings
minGain	Allows to adjust min gain setting to match individual card capabilities. Default range covers down to -4 dB
maxGain	Allows to adjust max gain setting to match individual card capabilities. Default range covers up to 45 dB
maxSlope	Allows to adjust max slope setting to match individual card capabilities. Default range covers up to 8 dB

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17)
config.monitoring	CHOICE	SETUP	EnumRange (YES NO)
config.hasLinearityMode	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasLowGainMode	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.minGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.maxGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.maxSlope	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
voltage.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
current.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
current.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
current.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
name	TEXT	R/O	StringRange R/O
linearityMode	CHOICE		EnumRange (STANDARD HIGH)
lowGainMode	CHOICE		EnumRange (ON OFF)
gain	FLOAT		DoubleRange (0.00 .. 28.00)
slope	INTEGER		IntegerRange (0 .. 10)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (VBUS/12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (3.3V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amp 1/2 Current Low)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amp 1/2 Current High)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amp 3/4 Current Low)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amp 3/4 Current High)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amp Temperature Low)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amp Temperature High)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Alto Temperature Low)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Alto Temperature High)

9.189 ETL-Genus-COM402

This page describes the *ETL-Genus-COM402* device driver and its device window. The*ETL-Genus-COM402* device driver monitors and controls one combiner card COM402 in a ETL Genus Chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:**COM-G3-S6-402**

- SW version: e599 1v5

Other available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lowLimit	Sets the lower rf power alarm limit in dBm.
highLimit	Sets the upper rf power alarm limit in dBm.
tripMonitoring	En- or disables the rf trip monitoring.
tripLimit	Sets the rf trip alarm limit in dBm.
targetLevel	Sets the rf target level in dBm.
rfTripAttn	Sets the rf trip attenuation.
slot	Selects the physical slot where the module is fitted in the chassis to control.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowLimit	INTEGER	SETUP	IntegerRange (-55 .. 5)
config.highLimit	INTEGER	SETUP	IntegerRange (-55 .. 5)
config.tripMonitoring	CHOICE	SETUP	EnumRange (ON OFF)
config.tripLimit	INTEGER	SETUP	IntegerRange (-30 .. 3)
config.targetLevel	FLOAT	SETUP	DoubleRange (-31.0 .. 3.0)
config.rfTripAttn	FLOAT	SETUP	DoubleRange (15.0 .. 30.0)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
voltage.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
location	CHOICE		EnumRange (1 2)
position	CHOICE	R/O	EnumRange R/O (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
gain	FLOAT		DoubleRange (0.00 .. 28.00)
slope	INTEGER		IntegerRange (0 .. 10)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (VBUS/12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Power Low)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RF Power High)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF Trip)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amp Communication)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Path A Amp 1-2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Path B Amp 1-2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Path A Amp 3-4)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Path B Amp 3-4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.190 ETL-Genus-DA-22512

This page describes the *ETL-Genus-DA-22512* device driver and its device window. The *ETL-Genus-DA-22512* device driver monitors and controls one dual input 16 port distribution amplifier in a ETL Genus Chassis.

The driver was tested with the following equipment:**DA-10MHz: M22512**

- SW version: e585 1v2

Other available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The device in and output rf alarm limits.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
mode	Selects the input signal type 10MHz DCLS IRIG IRIG AM
gainMode	Sets the units gain mode. AUTO LOW HIGH
default.Input	Sets the default input.
default.path	Sets the default amplifier path.
hotStandbyMode	En/Disables the units hot standby mode.
input.01.lowFreq	Sets the units input 1 lower frequency alarm limit.
input.01.highFreq	Sets the units input 1 upper frequency alarm limit.

parameter	description
input.02.lowFreq	Sets the units input 2 lower frequency alarm limit.
input.02.highFreq	Sets the units input 2 upper frequency alarm limit.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.frameVersion	TEXT	R/O	StringRange R/O
info.frameSerial	TEXT	R/O	StringRange R/O
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.12Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.opMode	CHOICE	SETUP	EnumRange (10MHz DCLS IRIG IRIG AM)
config.gainMode	CHOICE	SETUP	EnumRange (AUTO LOW HIGH)
config.default.input	CHOICE	SETUP	EnumRange (INPUT 1 INPUT 2)
config.default.path	CHOICE	SETUP	EnumRange (PATH A PATH B)
config.hotStandbyMode	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.input.01.lowFreq	FLOAT	SETUP	DoubleRange (0.000000 .. 0.000000)
config.input.01.highFreq	FLOAT	SETUP	DoubleRange (0.000000 .. 0.000000)
config.input.02.lowFreq	FLOAT	SETUP	DoubleRange (0.000000 .. 0.000000)
config.input.02.highFreq	FLOAT	SETUP	DoubleRange (0.000000 .. 0.000000)
internal.opMode	CHOICE	R/O	EnumRange R/O (10MHz DCLS IRIG IRIG AM)
id1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
protection	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
location	CHOICE		EnumRange (INPUT 1 INPUT 2)
position	CHOICE	R/O	EnumRange R/O (A B)
input.01.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.02.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.01.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
input.02.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
input.01.freq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
input.02.freq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
protection.1	CHOICE		EnumRange (AUTO MANUAL)
location.1	CHOICE		EnumRange (PATH A PATH B)
position.1	CHOICE	R/O	EnumRange R/O (A B)
gain	CHOICE	R/O	EnumRange R/O (LOW HIGH)
output.01.attn	INTEGER		IntegerRange (0 .. 10)
output.02.attn	INTEGER		IntegerRange (0 .. 10)
output.03.attn	INTEGER		IntegerRange (0 .. 10)
output.04.attn	INTEGER		IntegerRange (0 .. 10)
output.05.attn	INTEGER		IntegerRange (0 .. 10)
output.06.attn	INTEGER		IntegerRange (0 .. 10)
output.07.attn	INTEGER		IntegerRange (0 .. 10)
output.08.attn	INTEGER		IntegerRange (0 .. 10)
output.09.attn	INTEGER		IntegerRange (0 .. 10)
output.10.attn	INTEGER		IntegerRange (0 .. 10)
output.11.attn	INTEGER		IntegerRange (0 .. 10)

name	*type*	*flags*	*range*
output.12.attn	INTEGER		IntegerRange (0 .. 10)
output.13.attn	INTEGER		IntegerRange (0 .. 10)
output.14.attn	INTEGER		IntegerRange (0 .. 10)
output.15.attn	INTEGER		IntegerRange (0 .. 10)
output.16.attn	INTEGER		IntegerRange (0 .. 10)
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.01.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.02.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.03.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.04.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.05.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
output.06.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.07.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.08.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.09.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.10.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.11.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.12.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.13.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.14.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.15.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.16.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.01.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.02.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.03.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.04.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.05.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.06.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.07.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.08.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.09.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.10.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.11.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.12.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.13.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.14.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.15.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)
output.16.state	CHOICE	R/O	EnumRange R/O (OK LOW HIGH)

name	*type*	*flags*	*range*
amp.01.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.02.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.03.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.04.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.05.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.06.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.01.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.01.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.02.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
input.02.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.01.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.01.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.02.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.02.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.03.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.03.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.04.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.04.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.05.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.05.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.06.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.06.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.07.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.07.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.08.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)

name	*type*	*flags*	*range*
output.08.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.09.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.09.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.10.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.10.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.11.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.11.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.12.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.12.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.13.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.13.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.14.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.14.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.15.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.15.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.16.lowLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
output.16.highLimit	FLOAT		DoubleRange (-17.0 .. 25.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.04	ALARM	R/O	AlarmFlagRange R/O (12V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (3.3V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (-12V)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (CPU)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Input 1 RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input 2 RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Input 1 Freq)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input 2 Freq)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Amplifier 5)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Amplifier 6)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Input Protection)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Path Protection)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Output 1 RF/Sig)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Output 2 RF/Sig)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Output 3 RF/Sig)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Output 4 RF/Sig)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Output 5 RF/Sig)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Output 6 RF/Sig)

name	*type*	*flags*	*range*
faults.34	ALARM	R/O	AlarmFlagRange R/O (Output 7 RF/Sig)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Output 8 RF/Sig)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Output 9 RF/Sig)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Output 10 RF/Sig)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Output 11 RF/Sig)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Output 12 RF/Sig)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Output 13 RF/Sig)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Output 14 RF/Sig)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Output 15 RF/Sig)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Output 16 RF/Sig)

9.191 ETL-Genus-DIV401

This page describes the *ETL-Genus-DIV401* device driver and its device window. The *ETL-Genus-DIV401* device driver monitors and controls one splitter card DIV401 in a ETL Genus Chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following equipment: DIV-G3-S6-401

- SW version: e599 1v6

Other available optional cards of the ETL Genus series are:

- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lnbVoltage	Sets the lnb voltage if supported.
lnbTone	Sets the lnb tone/22kHz if supported.
lowLimit	Sets the lower rf power alarm limit in dBm.
highLimit	Sets the upper rf power alarm limit in dBm.
slot	Selects the physical slot where the module is fitted in the chassis to control.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.lnbTone	CHOICE	SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SETUP	IntegerRange (-55 .. 5)
config.highLimit	INTEGER	SETUP	IntegerRange (-55 .. 5)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
voltage.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.03	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
name	TEXT	R/O	StringRange R/O
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
location	CHOICE		EnumRange (1 2)
position	CHOICE	R/O	EnumRange R/O (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
gain	FLOAT		DoubleRange (0.00 .. 28.00)
slope	INTEGER		IntegerRange (0 .. 10)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (VBUS/12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LNB Voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LNB Current)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF Power Low)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Power High)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amp Communication)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Path A Amp 1-2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Path B Amp 1-2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Path A Amp 3-4)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Path B Amp 3-4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.192 ETL-Genus-Downconverter

This page describes the *ETL-Genus-Upconverter* device driver and its device window. The *ETL-Genus-Upconverter* device driver monitors and controls a Falcon series Upconverter module installed in a ETL Genus chassis.

The driver was tested with the following equipment:

- Genus Frame: GNS-106-1U
  - SW version: e530 2v24

and Upconverter model:

- FN-U-L1F2-24405AA
  - SW version: e719 1v13

Other available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module
- ETL-Genus-10MHz monitors and controls one OSC 10MHz reference module
- ETL-Genus-100MHz monitors and controls one OSC 100MHz reference module
- ETL-Genus-Downconverter monitors and controls one Falcon series downconverter module

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The converter settings.
- --- The converter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section
setLimits	En- or Disables Limit Alarm Monitoring
rf.loLimit	Sets the low rf limit
rf.ihLImit	Sets the high rf limit
minGain	Defines the minimum available gain of the converter
maxGain	Defines the maximum available gain of the converter
hasSlope	Defines is the unit is equipped with a adjustable slope compensator
slot	Defines the table slot the converter is installed within the chassis
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.v3	TEXT	R/O	StringRange R/O
internal.present	TEXT	R/O	StringRange R/O
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.range	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.rf.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.rf.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.minGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.maxGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hasSlope	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
refPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.on	CHOICE		EnumRange (ON OFF)
rx.gain	FLOAT		DoubleRange (5.0 .. 35.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.slope	INTEGER		IntegerRange DIS (0 .. 20)
rx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.frequency	CHOICE		EnumRange (70 MHz 140 MHz)

name	*type*	*flags*	*range*
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input Lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output Lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF1 3.3V Amp V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF2 3.3V Amp V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (RF1 5V Amp I)
faults.09	ALARM	R/O	AlarmFlagRange R/O (RF2 5V Amp I)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RF3 Amp I)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RF4 Amp I)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF1 3.3V Amp I)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF2 3.3V Amp I)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RF Power)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reference Monitor)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX off)

9.193 ETL-Genus-Frame

This page describes the *ETL-Genus-Frame* device driver and its device window. The *ETL-Genus-Frame* device driver monitors and controls the base frame/unit features and alarms of one ETL Genus chassis. The frame driver supports up to 16 modules and allows to set each modules alias name as well as reset the whole frame.

Beside this base unit multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap. Single module units (e.g. 22512 16 Port Distribution Amplifier) are handled within one separate driver. These units do not require an additional Base Frame device instance.

The driver was tested with the following equipment:

- *CPU: CPU0086*
  - SW version: e530 1v7
- *PSU: MPS1170*
  - SW version: e567 1v9

The different available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules alias names and PSU readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
noModules	Defines the number of modules installed in the main frame (0-16).
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.noModules	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.hasFan	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
module.01.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.02.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.03.enable	BOOLEAN	SAVE	BooleanRange (true false)

name	*type*	*flags*	*range*
module.04.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.05.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.06.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.07.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.08.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.09.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.10.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.11.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.12.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.13.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.14.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.15.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.16.enable	BOOLEAN	SAVE	BooleanRange (true false)
module.01.type	TEXT	R/O	StringRange R/O
module.02.type	TEXT	R/O	StringRange R/O
module.03.type	TEXT	R/O	StringRange R/O
module.04.type	TEXT	R/O	StringRange R/O
module.05.type	TEXT	R/O	StringRange R/O
module.06.type	TEXT	R/O	StringRange R/O
module.07.type	TEXT	R/O	StringRange R/O
module.08.type	TEXT	R/O	StringRange R/O
module.09.type	TEXT	R/O	StringRange R/O
module.10.type	TEXT	R/O	StringRange R/O
module.11.type	TEXT	R/O	StringRange R/O
module.12.type	TEXT	R/O	StringRange R/O
module.13.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.14.type	TEXT	R/O	StringRange R/O
module.15.type	TEXT	R/O	StringRange R/O
module.16.type	TEXT	R/O	StringRange R/O
module.01.name	TEXT		StringRange
module.02.name	TEXT		StringRange
module.03.name	TEXT		StringRange
module.04.name	TEXT		StringRange
module.05.name	TEXT		StringRange
module.06.name	TEXT		StringRange
module.07.name	TEXT		StringRange
module.08.name	TEXT		StringRange
module.09.name	TEXT		StringRange
module.10.name	TEXT		StringRange
module.11.name	TEXT		StringRange
module.12.name	TEXT		StringRange
module.13.name	TEXT		StringRange
module.14.name	TEXT		StringRange
module.15.name	TEXT		StringRange
module.16.name	TEXT		StringRange
psu.1.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.1.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
fan.2.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.3.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.4.speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clear	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.psuComms	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.psuLine	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (MODULE 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (MODULE 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (MODULE 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MODULE 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (MODULE 5)
faults.08	ALARM	R/O	AlarmFlagRange R/O (MODULE 6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (MODULE 7)
faults.10	ALARM	R/O	AlarmFlagRange R/O (MODULE 8)
faults.11	ALARM	R/O	AlarmFlagRange R/O (MODULE 9)
faults.12	ALARM	R/O	AlarmFlagRange R/O (MODULE 10)
faults.13	ALARM	R/O	AlarmFlagRange R/O (MODULE 11)
faults.14	ALARM	R/O	AlarmFlagRange R/O (MODULE 12)
faults.15	ALARM	R/O	AlarmFlagRange R/O (MODULE 13)
faults.16	ALARM	R/O	AlarmFlagRange R/O (MODULE 14)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (MODULE 15)
faults.18	ALARM	R/O	AlarmFlagRange R/O (MODULE 16)
faults.19	ALARM	R/O	AlarmFlagRange R/O (CPU)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Fan Summary)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Fan 4)

9.194 ETL-Genus-PRN

This page describes the *ETL-Genus-PRN* device driver and its device window. The *ETL-Genus-PRN* device driver monitors and controls one Piranha LNB or BUC DC Injection Card in a Genus Chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following equipment:

- Genus Frame 196-2
  - SW version: e530 2v19

and card models:

- PRN-G1S-LS2-106
  - SW version: e620 1v22
- PRN-G1S-BS2-003
  - SW version: e620 1v22

The different available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module

- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
voltage	Sets the DC supply voltage for the individual card. Depending on the card the following voltages are available 13V/15V/18V/48V
22kHz	En- or Disable the 22kHz tone
reference	En- or Disables the reference usage
current.loLimit	Sets the lower supply current alarm limit
current.hiLimit	Sets the upper supply current alarm limit
rfLevel.loLimit	Sets the lower rf level alarm limit
rfLevel.hiLimit	Sets the upper rf level alarm limit
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.vBus	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.5v	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.elapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.voltage	INTEGER	SETUP	IntegerRange (0 .. 0)
config.22kHz	CHOICE	SETUP	EnumRange (ON OFF)
config.current.loLimit	INTEGER	SETUP	IntegerRange (0 .. 6000)
config.current.hiLimit	INTEGER	SETUP	IntegerRange (0 .. 6000)
config.rfLevel.loLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.rfLevel.hiLimit	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Voltage Low)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Voltage High)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Current Low)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Current High)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF Power Low)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Power High)
faults.08	ALARM	R/O	AlarmFlagRange R/O (VBus)
faults.09	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Voltage Off)

9.195 ETL-Genus-SRY-RX

This page describes the *ETL-Genus-SRY-RX* device driver and its device window. The *ETL-Genus-SRY-RX* device driver monitors and controls one Stingray Optical Receiver Card in a ETL Genus chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following card models:

- SRY-G1S-RS6-162
  - SW version: e592 1v6

The different available optional cards of the ETL Genus series are:

- ETL-Genus-SRY-TX monitors and controls one fo stingray transmit module
- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-ALT monitors and controls one amplifier card ALT series

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35
setLimits	En- or Disables Alarm Monitoring
output.loLimit	Sets the lower output level alarm limit
output.hiLimit	Sets the upper output alarm level limit
input.loLimit	Sets the lower input level alarm limit
input.hiLimit	Sets the upper input level alarm limit
reference	En- or Disables the 10MHz reference
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

parameter	description
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.VBusVolt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.output.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.output.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.input.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.input.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.reference	CHOICE	SETUP	EnumRange (OFF ON)
config.refPowerSet	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.gain	INTEGER		IntegerRange (0 .. 60)
rx.agc	CHOICE		EnumRange (AGC MANUAL FIXED)
rfPowerSet	FLOAT		DoubleRange (0.00 .. 0.00)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RF Output)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Optical Input)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)

9.196 ETL-Genus-SRY-TX

This page describes the *ETL-Genus-SRY-TX* device driver and its device window. The *ETL-Genus-SRY-TX* device driver monitors and controls one Stingray Optical Transmitter Card in a ETL Genus chassis.

Multiple instances of the device may be configured in the sat-nms M&C to handle all modules of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical slots in the Genus chassis this particular sat-nms device shall control.

The driver was tested with the following card models:

- SRY-G1S-TS6-161
  - SW version: e583 1v7

The different available optional cards of the ETL Genus series are:

- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-ALT monitors and controls one amplifier card ALT series

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modules settings and readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35

parameter	description
setLimits	En- or Disables Alarm Monitoring
output.loLimit	Sets the lower output level alarm limit
output.hiLimit	Sets the upper output alarm level limit
input.loLimit	Sets the lower input level alarm limit
input.hiLimit	Sets the upper input level alarm limit
voltage	Sets the lnb supply voltage
22kHz	En- or Disables the lnb 22KHz tone signal
lnb.loLimit	Sets the lower lnb supply current alarm limit
lnb.hiLimit	Sets the upper lnb supply current alarms limit
reference	En- or Disables the 10MHz reference
slot	Defines the table index the 10MHz modules is installed within the chassis. Default value is 35
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.VBusVolt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.output.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.output.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.input.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.input.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.voltage	INTEGER	SETUP	IntegerRange (0 .. 0)
config.22kHz	CHOICE	SETUP	EnumRange (ON OFF)
config.lnb.loCurrent	INTEGER	SETUP	IntegerRange (0 .. 0)
config.lnb.hiCurrent	INTEGER	SETUP	IntegerRange (0 .. 0)
config.reference	CHOICE	SETUP	EnumRange (OFF ON)
config.refPowerSet	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rfOutPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.gain	INTEGER		IntegerRange (0 .. 60)
tx.agc	CHOICE		EnumRange (AGC MANUAL FIXED)
rfPowerSet	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPowerSet	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RF Input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Optical Output)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (LNB Voltage)
faults.08	ALARM	R/O	AlarmFlagRange R/O (LNB Current)

9.197 ETL-Genus-Upconverter

This page describes the *ETL-Genus-Downconverter* device driver and its device window. The *ETL-Genus-Downconverter* device driver monitors and controls a Falcon series Downconverter module installed in a ETL Genus chassis.

The driver was tested with the following equipment:

Genus Frame: GNS-106-1U

- SW version:
- e530 2v20
- e530 2v24

and Downconverter model: FN-D-L1F2-24404AA

- SW version:
- e719 1v3
- e719 1v13

Other available optional cards of the ETL Genus series are:

- ETL-Genus-DIV401 monitors and controls one splitter module DIV 401
- ETL-Genus-COM402 monitors and controls one combiner module COM 402
- ETL-Genus-DA-22512 monitors and controls one 10MHz Distribution amplifier M22512
- ETL-Genus-ALT monitors and controls one amplifier card ALT series
- ETL-Genus-PRN monitors and controls one Piranha LNB or BUC DC Injection Card.
- ETL-Genus-SRY-RX monitors and controls one fo stingray receiver module
- ETL-Genus-SRY-TX monitors and controls one fo stingray transmitter module
- ETL-Genus-10MHz monitors and controls one OSC 10MHz reference module
- ETL-Genus-100MHz monitors and controls one OSC 100MHz reference module
- ETL-Genus-Upconverter monitors and controls one Falcon series upconverter module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The converter settings.
- --- The converter readings.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section
setLimits	En- or Disables Limit Alarm Monitoring
rf.loLimit	Sets the low rf limit
rf.ihLImit	Sets the high rf limit
minGain	Defines the minimum available gain of the converter
maxGain	Defines the maximum available gain of the converter
hasSlope	Defines is the unit is equipped with a adjustable slope compensator
slot	Defines the slot the converter is installed within the chassis
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.

parameter	description
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.present	TEXT	R/O	StringRange R/O
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.range	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.rf.loLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.rf.hiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.minGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.maxGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hasSlope	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rfPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
refPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.on	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (5.0 .. 25.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.slope	INTEGER		IntegerRange DIS (0 .. 20)
rx.frequency	CHOICE		EnumRange (70 MHz 140 MHz)
tx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input Lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output Lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF1 3.3V Amp V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF2 3.3V Amp V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (RF1 5V Amp I)
faults.09	ALARM	R/O	AlarmFlagRange R/O (RF2 5V Amp I)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RF 9V Amp I)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RF4 Amp I)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF1 3.3V Amp I)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF2 3.3V Amp I)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RF Power)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reference Monitor)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX off)

9.198 ETL-Genus-VTR-101-Matrix

This page describes the *ETL-Genus-VTR-101-Matrix* device driver and its device window. The *ETL-Genus-VTR-101-Matrix* device driver monitors and controls one distributive VTR-101 matrix in a genus chassis with up to 16 inputs and 16 outputs.

As the number of installed in and outputs can not be read from the device, it is necessary to configure the matrix dimensions within the software using the device configuration page.

The driver was tested with the following equipment:

- VTR-101 16x16
  - SW version: e672 1v13

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by alias name
- 1 2 3 --- Port switching by number
- --- The amplifier readings
- --- Edit in and output alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	**description**

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
chassisMon	En- or Disables Alarm Monitoring
inputs	Defines the number of installed inputs
outputs	Defines the number of installed outputs
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.matrixCpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rfTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltageNeg5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.chassisMon	CHOICE	SETUP	EnumRange (ON OFF)
config.inputs	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
config.outputs	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
aliasRead	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
input.01.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.03.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.04.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.05.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.06.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.07.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.08.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.09.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.10.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
input.11.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.12.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.13.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.14.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.15.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.16.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.01.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.03.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.04.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.05.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.06.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.07.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.08.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.09.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.10.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.11.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.12.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.13.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.14.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.15.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.16.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.01.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.02.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.03.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.04.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.05.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.06.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.07.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.08.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.09.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.10.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.11.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.12.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.13.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.14.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.15.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.16.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
output.01.locking	INTEGER		IntegerRange (0 .. 1)
output.02.locking	INTEGER		IntegerRange (0 .. 1)
output.03.locking	INTEGER		IntegerRange (0 .. 1)
output.04.locking	INTEGER		IntegerRange (0 .. 1)
output.05.locking	INTEGER		IntegerRange (0 .. 1)
output.06.locking	INTEGER		IntegerRange (0 .. 1)
output.07.locking	INTEGER		IntegerRange (0 .. 1)
output.08.locking	INTEGER		IntegerRange (0 .. 1)
output.09.locking	INTEGER		IntegerRange (0 .. 1)
output.10.locking	INTEGER		IntegerRange (0 .. 1)
output.11.locking	INTEGER		IntegerRange (0 .. 1)
output.12.locking	INTEGER		IntegerRange (0 .. 1)
output.13.locking	INTEGER		IntegerRange (0 .. 1)
output.14.locking	INTEGER		IntegerRange (0 .. 1)
output.15.locking	INTEGER		IntegerRange (0 .. 1)
output.16.locking	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU +5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU -5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (CPU Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Matrix CPU Temperature)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input 9)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input 15)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Input 16)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Output 9)

name	*type*	*flags*	*range*
faults.40	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Output 15)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Output 16)

9.199 ETL-Genus-VTRC-101-Matrix

This page describes the *ETL-Genus-VTR-101C-Matrix* device driver and its device window. The *ETL-Genus-VTR-101C-Matrix* device driver monitors and controls one combining VTR-101 matrix in a genus chassis with up to 16 inputs and 16 outputs.

As the number of installed in and outputs can not be read from the device, it is necessary to configure the matrix dimensions within the software using the device configuration page.

The driver was tested with the following equipment:

- VTR-101C 16x16
  - SW version: e672 1v13

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching by alias name
- --- Port switching by number
- --- The amplifier readings
- --- Edit in and output alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
chassisMon	En- or Disables Alarm Monitoring
inputs	Defines the number of installed inputs
outputs	Defines the number of installed outputs
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.matrixCpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rfTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltageNeg5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.chassisMon	CHOICE	SETUP	EnumRange (ON OFF)
config.inputs	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
config.outputs	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.deviceComment	TEXT	SAVE SETUP	StringRange
aliasRead	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.01	CHOICE		EnumRange (OFF)
input.02	CHOICE		EnumRange (OFF)
input.03	CHOICE		EnumRange (OFF)
input.04	CHOICE		EnumRange (OFF)
input.05	CHOICE		EnumRange (OFF)
input.06	CHOICE		EnumRange (OFF)
input.07	CHOICE		EnumRange (OFF)
input.08	CHOICE		EnumRange (OFF)
input.09	CHOICE		EnumRange (OFF)
input.10	CHOICE		EnumRange (OFF)
input.11	CHOICE		EnumRange (OFF)
input.12	CHOICE		EnumRange (OFF)
input.13	CHOICE		EnumRange (OFF)
input.14	CHOICE		EnumRange (OFF)
input.15	CHOICE		EnumRange (OFF)
input.16	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
output.01.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.02.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.03.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.04.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.05.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.06.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.07.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.08.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.09.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.10.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.11.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.12.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.13.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.14.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.15.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.16.amp1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.01.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.02.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.03.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.04.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.05.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.06.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.07.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.08.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.09.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.10.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.11.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.12.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.13.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.14.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.15.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.16.amp2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.01.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.03.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.04.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
input.05.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.06.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.07.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.08.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.09.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.10.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.11.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.12.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.13.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.14.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.15.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.16.ampCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
input.01.locking	INTEGER		IntegerRange (0 .. 1)
input.02.locking	INTEGER		IntegerRange (0 .. 1)
input.03.locking	INTEGER		IntegerRange (0 .. 1)
input.04.locking	INTEGER		IntegerRange (0 .. 1)
input.05.locking	INTEGER		IntegerRange (0 .. 1)
input.06.locking	INTEGER		IntegerRange (0 .. 1)
input.07.locking	INTEGER		IntegerRange (0 .. 1)
input.08.locking	INTEGER		IntegerRange (0 .. 1)
input.09.locking	INTEGER		IntegerRange (0 .. 1)
input.10.locking	INTEGER		IntegerRange (0 .. 1)
input.11.locking	INTEGER		IntegerRange (0 .. 1)
input.12.locking	INTEGER		IntegerRange (0 .. 1)
input.13.locking	INTEGER		IntegerRange (0 .. 1)
input.14.locking	INTEGER		IntegerRange (0 .. 1)
input.15.locking	INTEGER		IntegerRange (0 .. 1)
input.16.locking	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU +5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU -5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (CPU Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Matrix CPU Temperature)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Output 15)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Output 16)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Input 9)

name	*type*	*flags*	*range*
faults.40	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Input 15)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Input 16)

9.200 ETL-HMI-Reboot

This page describes the device driver and the device window for the ETL-HMI-Reboot device. This device provides a "REBOOT" button for user interface panels of ETL matrices, permits to reboot such a device when the button is clicked.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window the IP address of the device to reboot must be set as well as user name and password to login at the device via SSH.

parameter	*description*

parameter	*description*
address	The device address expected by this protocol either is a known host name or a IP4 address in dotted quad notation, optionally followed by colon and a port number (the protocol uses port 22 if no port is specified). This must be followed by the user name and password to log in at the SSH account where the command shall be executed. Example: 10.10.1.1:22 user password Optionally the password may be followed by an integer number which is interpreted as the connection / reply timeout (msecs) for the ssh command. Default timeout is 2000msecs.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
reboot	TEXT	NOPRESET	StringRange

9.201 ETL-Hurricane-Matrix

This page describes the device driver and the device window for ETL Hurricane L-Band switch matrix series. The driver works with a single 64x64 port matrix of the series.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name
- 123 --- Port switching by number

- --- Edit input names.
- --- Edit output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the READ ALIAS NAMES FROM CHASSIS button on the device driver. The reason is that reading these value is only done initially at the server startup due to the slow device SNMP performance.

When editing port alias names through the sat-nms software, please note that the device does not accept alias names containing other characters than A-Z (upper and lower case), 0-9 and the hyphen character '-'. If you enter alias names which contain unsupported characters, the device does not change the name.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.outputs	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
aliasRead	TEXT		StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
output.33.name	TEXT		StringRange
output.34.name	TEXT		StringRange
output.35.name	TEXT		StringRange
output.36.name	TEXT		StringRange
output.37.name	TEXT		StringRange
output.38.name	TEXT		StringRange
output.39.name	TEXT		StringRange
output.40.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.41.name	TEXT		StringRange
output.42.name	TEXT		StringRange
output.43.name	TEXT		StringRange
output.44.name	TEXT		StringRange
output.45.name	TEXT		StringRange
output.46.name	TEXT		StringRange
output.47.name	TEXT		StringRange
output.48.name	TEXT		StringRange
output.49.name	TEXT		StringRange
output.50.name	TEXT		StringRange
output.51.name	TEXT		StringRange
output.52.name	TEXT		StringRange
output.53.name	TEXT		StringRange
output.54.name	TEXT		StringRange
output.55.name	TEXT		StringRange
output.56.name	TEXT		StringRange
output.57.name	TEXT		StringRange
output.58.name	TEXT		StringRange
output.59.name	TEXT		StringRange
output.60.name	TEXT		StringRange
output.61.name	TEXT		StringRange
output.62.name	TEXT		StringRange
output.63.name	TEXT		StringRange
output.64.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange
input.43.name	TEXT		StringRange
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.55.name	TEXT		StringRange
input.56.name	TEXT		StringRange
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.02	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.03	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.04	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.05	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.06	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.07	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.08	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.09	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.10	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.11	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.12	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.13	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.14	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.15	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.16	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.17	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.18	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.19	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.20	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.21	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.22	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.23	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.24	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.25	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.26	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.27	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.28	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.29	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.30	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.31	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.32	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.33	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.34	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.35	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.36	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.37	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.38	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.39	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.40	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.41	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.42	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.43	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.44	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.45	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.46	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.47	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.48	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.49	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.50	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.51	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.52	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.53	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.54	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.55	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.56	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.57	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.58	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.59	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.60	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.61	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.62	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.63	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.64	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.01	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.02	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.03	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.04	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.05	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.06	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.07	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.08	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.09	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.10	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.11	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.12	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.13	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.14	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.15	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.16	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.17	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.18	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.19	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.20	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.21	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.22	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.23	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.24	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.25	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.26	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.27	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.28	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.29	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.30	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.31	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.32	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.33	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.34	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.35	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.36	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.37	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.38	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.39	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.40	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.41	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.42	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.43	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.44	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.45	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.46	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.47	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.48	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.49	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.50	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.51	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.52	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.53	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.54	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.55	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.56	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.57	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.58	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.59	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.60	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
byname.61	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.62	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.63	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
byname.64	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.202 ETL-LS-Switch

This page describes the device driver and the device window for etl systems LS Switch with 8, 16 or 32 inputs.

To allow proper operation the number of inputs has to be defined in the configuration page. Please note: With the tested SW releases 1.3 and 1.6 the auto save and remote mode doesn't work properly. Therefore they are not part of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device including the number of inputs.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- inputs --- Defines number of inputs. This is necessary for proper communication.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.unitTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.inputs	CHOICE	SAVE SETUP	EnumRange (8 16 32)
position	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
byname	CHOICE		EnumRange (0-Terminate 01 02 03 04 05 06 07 08)
reInit	TEXT	NOPRESET	StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange DIS
input.10.name	TEXT		StringRange DIS
input.11.name	TEXT		StringRange DIS

name	*type*	*flags*	*range*
input.12.name	TEXT		StringRange DIS
input.13.name	TEXT		StringRange DIS
input.14.name	TEXT		StringRange DIS
input.15.name	TEXT		StringRange DIS
input.16.name	TEXT		StringRange DIS
input.17.name	TEXT		StringRange DIS
input.18.name	TEXT		StringRange DIS
input.19.name	TEXT		StringRange DIS
input.20.name	TEXT		StringRange DIS
input.21.name	TEXT		StringRange DIS
input.22.name	TEXT		StringRange DIS
input.23.name	TEXT		StringRange DIS
input.24.name	TEXT		StringRange DIS
input.25.name	TEXT		StringRange DIS
input.26.name	TEXT		StringRange DIS
input.27.name	TEXT		StringRange DIS
input.28.name	TEXT		StringRange DIS
input.29.name	TEXT		StringRange DIS
input.30.name	TEXT		StringRange DIS
input.31.name	TEXT		StringRange DIS
input.32.name	TEXT		StringRange DIS
internal.choices	TEXT	R/O	StringRange R/O
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.posName	TEXT	R/O	StringRange R/O
internal.init	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

name	*type*	*flags*	*range*

9.203 ETL-MatrixPort-Emulator

The *ETL-MatrixPort-Emulator* logical device permits to control one input/output port of a switch matrix using the same communication protocol as the *ETL-PDSwitch-Emulator*. It listens at a configurable IP port and uses this simplified MiteqMOD95 protocol to control one matrix port variable from an external software. The 2-digit matrix port position is transferred without change in both directions.

The device unterstands two commands to query the actual status and to set the matrix port to a certain position:

- Sending `{A?}` to the device queries the actual port setting. The device responds with `{A,nn}` where *nn* is the number of the actually selected port (2-digit, leading zero).*00* means no port is selected, *01* port number 1 and so on.
- Sending `{ACnn}` to the device commands a new port setting. The device responds with `{A,nn}` where *nn* is the number of the actually selected port (2-digit, leading zero).*00* means no port is selected, *01* port number 1 and so on.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- Shows the device's faults.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the *ETL-PDSwitch-Emulator* device with the device it shall control.

parameter	*description*
portNo	This configures the port number the device is listening. The valid range is 1024 .. 65535. You cannot use a port number that is already used by another application. The device will show a fault if it cannot open the port because it is already in use.
outputId	This is the message identifier of the output (or input) to control.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.indication	TEXT	R/O	StringRange R/O
config.portNo	INTEGER	SAVE SETUP	IntegerRange (1024 .. 65535)
config.outputId	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IP Port)

9.204 ETL-Modular-Frame-26128

This page describes the device driver and the device window for etl systems modular frame 26128.

The installed modules have to be addressed individually. To allow proper operation the serial device address has to be defined in the configuration page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Main frame.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	Must always be set to B if the device is controlled TCP or over a serial line.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
module.01.present	TEXT	R/O	StringRange R/O
module.02.present	TEXT	R/O	StringRange R/O
module.03.present	TEXT	R/O	StringRange R/O
module.04.present	TEXT	R/O	StringRange R/O
module.05.present	TEXT	R/O	StringRange R/O
module.06.present	TEXT	R/O	StringRange R/O
module.07.present	TEXT	R/O	StringRange R/O
module.08.present	TEXT	R/O	StringRange R/O
module.09.present	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.10.present	TEXT	R/O	StringRange R/O
module.11.present	TEXT	R/O	StringRange R/O
module.12.present	TEXT	R/O	StringRange R/O
module.13.present	TEXT	R/O	StringRange R/O
module.14.present	TEXT	R/O	StringRange R/O
module.15.present	TEXT	R/O	StringRange R/O
module.16.present	TEXT	R/O	StringRange R/O
module.01.type	TEXT	R/O	StringRange R/O
module.02.type	TEXT	R/O	StringRange R/O
module.03.type	TEXT	R/O	StringRange R/O
module.04.type	TEXT	R/O	StringRange R/O
module.05.type	TEXT	R/O	StringRange R/O
module.06.type	TEXT	R/O	StringRange R/O
module.07.type	TEXT	R/O	StringRange R/O
module.08.type	TEXT	R/O	StringRange R/O
module.09.type	TEXT	R/O	StringRange R/O
module.10.type	TEXT	R/O	StringRange R/O
module.11.type	TEXT	R/O	StringRange R/O
module.12.type	TEXT	R/O	StringRange R/O
module.13.type	TEXT	R/O	StringRange R/O
module.14.type	TEXT	R/O	StringRange R/O
module.15.type	TEXT	R/O	StringRange R/O
module.16.type	TEXT	R/O	StringRange R/O
psu.1.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
psu.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (MODULE 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (MODULE 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (MODULE 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MODULE 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (MODULE 5)
faults.08	ALARM	R/O	AlarmFlagRange R/O (MODULE 6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (MODULE 7)
faults.10	ALARM	R/O	AlarmFlagRange R/O (MODULE 8)
faults.11	ALARM	R/O	AlarmFlagRange R/O (MODULE 9)
faults.12	ALARM	R/O	AlarmFlagRange R/O (MODULE 10)
faults.13	ALARM	R/O	AlarmFlagRange R/O (MODULE 11)
faults.14	ALARM	R/O	AlarmFlagRange R/O (MODULE 12)
faults.15	ALARM	R/O	AlarmFlagRange R/O (MODULE 13)
faults.16	ALARM	R/O	AlarmFlagRange R/O (MODULE 14)
faults.17	ALARM	R/O	AlarmFlagRange R/O (MODULE 15)
faults.18	ALARM	R/O	AlarmFlagRange R/O (MODULE 16)

9.205 ETL-NGM100-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix NGM-100 series.

The driver works with the distributive matrix only. It supports 32x32 and 64x32 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-32.
- abc --- Port switching by name 33-64.
- 1 2 3 --- Port switching by number 1-32.
- 1 2 3 --- Port switching by number 33-64.
- --- Edit input names.
- --- Edit output names.
- --- Redundancy settings.
- --- External input fault configuration.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- address --- The SNMP address specification for the device. The full address specification consists of the following fields: *aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the READ ALIAS NAMES FROM CHASSIS button on the device driver. The reason is that reading these value is only done initially at the server startup due to the slow device SNMP performance.

External Input Faults

When checking the configtion for a input card redundancy switching, the driver not only checks the input card faults, but also an user configurable external fault for every matrix input. This external fault for instance may monitor the health of an fibre optics receiver in front of the

matrix input. The IDs for these externals faults are configured at a separate page of the device screen. Please note, that because the IDs are considered as configuration parameters, a privilege level of 150 is required to change these.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inhibited	TEXT	R/O	StringRange R/O
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu3Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu4Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu3FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu4FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.comFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.psuFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.inhibitId	TEXT	SAVE SETUP	StringRange
config.inhibitVal	TEXT	SAVE SETUP	StringRange
info.version	TEXT	R/O	StringRange R/O
info.swReleaseDate	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
protection.inhibit	BOOLEAN	NOPRESET	BooleanRange (true false)
aliasRead	TEXT		StringRange
bulk	TEXT	NOPRESET	StringRange
input.01.present	CHOICE	R/O	EnumRange R/O (0 1)
input.02.present	CHOICE	R/O	EnumRange R/O (0 1)
input.03.present	CHOICE	R/O	EnumRange R/O (0 1)
input.04.present	CHOICE	R/O	EnumRange R/O (0 1)
input.05.present	CHOICE	R/O	EnumRange R/O (0 1)
input.06.present	CHOICE	R/O	EnumRange R/O (0 1)
input.07.present	CHOICE	R/O	EnumRange R/O (0 1)
input.08.present	CHOICE	R/O	EnumRange R/O (0 1)
input.09.present	CHOICE	R/O	EnumRange R/O (0 1)
input.10.present	CHOICE	R/O	EnumRange R/O (0 1)
input.11.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.12.present	CHOICE	R/O	EnumRange R/O (0 1)
input.13.present	CHOICE	R/O	EnumRange R/O (0 1)
input.14.present	CHOICE	R/O	EnumRange R/O (0 1)
input.15.present	CHOICE	R/O	EnumRange R/O (0 1)
input.16.present	CHOICE	R/O	EnumRange R/O (0 1)
input.17.present	CHOICE	R/O	EnumRange R/O (0 1)
input.18.present	CHOICE	R/O	EnumRange R/O (0 1)
input.19.present	CHOICE	R/O	EnumRange R/O (0 1)
input.20.present	CHOICE	R/O	EnumRange R/O (0 1)
input.21.present	CHOICE	R/O	EnumRange R/O (0 1)
input.22.present	CHOICE	R/O	EnumRange R/O (0 1)
input.23.present	CHOICE	R/O	EnumRange R/O (0 1)
input.24.present	CHOICE	R/O	EnumRange R/O (0 1)
input.25.present	CHOICE	R/O	EnumRange R/O (0 1)
input.26.present	CHOICE	R/O	EnumRange R/O (0 1)
input.27.present	CHOICE	R/O	EnumRange R/O (0 1)
input.28.present	CHOICE	R/O	EnumRange R/O (0 1)
input.29.present	CHOICE	R/O	EnumRange R/O (0 1)
input.30.present	CHOICE	R/O	EnumRange R/O (0 1)
input.31.present	CHOICE	R/O	EnumRange R/O (0 1)
input.32.present	CHOICE	R/O	EnumRange R/O (0 1)
input.33.present	CHOICE	R/O	EnumRange R/O (0 1)
input.34.present	CHOICE	R/O	EnumRange R/O (0 1)
input.35.present	CHOICE	R/O	EnumRange R/O (0 1)
input.36.present	CHOICE	R/O	EnumRange R/O (0 1)
input.37.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.38.present	CHOICE	R/O	EnumRange R/O (0 1)
input.39.present	CHOICE	R/O	EnumRange R/O (0 1)
input.40.present	CHOICE	R/O	EnumRange R/O (0 1)
input.41.present	CHOICE	R/O	EnumRange R/O (0 1)
input.42.present	CHOICE	R/O	EnumRange R/O (0 1)
input.43.present	CHOICE	R/O	EnumRange R/O (0 1)
input.44.present	CHOICE	R/O	EnumRange R/O (0 1)
input.45.present	CHOICE	R/O	EnumRange R/O (0 1)
input.46.present	CHOICE	R/O	EnumRange R/O (0 1)
input.47.present	CHOICE	R/O	EnumRange R/O (0 1)
input.48.present	CHOICE	R/O	EnumRange R/O (0 1)
input.49.present	CHOICE	R/O	EnumRange R/O (0 1)
input.50.present	CHOICE	R/O	EnumRange R/O (0 1)
input.51.present	CHOICE	R/O	EnumRange R/O (0 1)
input.52.present	CHOICE	R/O	EnumRange R/O (0 1)
input.53.present	CHOICE	R/O	EnumRange R/O (0 1)
input.54.present	CHOICE	R/O	EnumRange R/O (0 1)
input.55.present	CHOICE	R/O	EnumRange R/O (0 1)
input.56.present	CHOICE	R/O	EnumRange R/O (0 1)
input.57.present	CHOICE	R/O	EnumRange R/O (0 1)
input.58.present	CHOICE	R/O	EnumRange R/O (0 1)
input.59.present	CHOICE	R/O	EnumRange R/O (0 1)
input.60.present	CHOICE	R/O	EnumRange R/O (0 1)
input.61.present	CHOICE	R/O	EnumRange R/O (0 1)
input.62.present	CHOICE	R/O	EnumRange R/O (0 1)
input.63.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.64.present	CHOICE	R/O	EnumRange R/O (0 1)
input.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)
input.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
input.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)
input.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
input.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
input.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
input.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)
input.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)
input.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)
input.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
input.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)
input.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
input.01.faultId	TEXT	SAVE	StringRange
input.02.faultId	TEXT	SAVE	StringRange
input.03.faultId	TEXT	SAVE	StringRange
input.04.faultId	TEXT	SAVE	StringRange
input.05.faultId	TEXT	SAVE	StringRange
input.06.faultId	TEXT	SAVE	StringRange
input.07.faultId	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
input.08.faultId	TEXT	SAVE	StringRange
input.09.faultId	TEXT	SAVE	StringRange
input.10.faultId	TEXT	SAVE	StringRange
input.11.faultId	TEXT	SAVE	StringRange
input.12.faultId	TEXT	SAVE	StringRange
input.13.faultId	TEXT	SAVE	StringRange
input.14.faultId	TEXT	SAVE	StringRange
input.15.faultId	TEXT	SAVE	StringRange
input.16.faultId	TEXT	SAVE	StringRange
input.17.faultId	TEXT	SAVE	StringRange
input.18.faultId	TEXT	SAVE	StringRange
input.19.faultId	TEXT	SAVE	StringRange
input.20.faultId	TEXT	SAVE	StringRange
input.21.faultId	TEXT	SAVE	StringRange
input.22.faultId	TEXT	SAVE	StringRange
input.23.faultId	TEXT	SAVE	StringRange
input.24.faultId	TEXT	SAVE	StringRange
input.25.faultId	TEXT	SAVE	StringRange
input.26.faultId	TEXT	SAVE	StringRange
input.27.faultId	TEXT	SAVE	StringRange
input.28.faultId	TEXT	SAVE	StringRange
input.29.faultId	TEXT	SAVE	StringRange
input.30.faultId	TEXT	SAVE	StringRange
input.31.faultId	TEXT	SAVE	StringRange
input.32.faultId	TEXT	SAVE	StringRange
output.01.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
output.33.name	TEXT		StringRange
output.34.name	TEXT		StringRange
output.35.name	TEXT		StringRange
output.36.name	TEXT		StringRange
output.37.name	TEXT		StringRange
output.38.name	TEXT		StringRange
output.39.name	TEXT		StringRange
output.40.name	TEXT		StringRange
output.41.name	TEXT		StringRange
output.42.name	TEXT		StringRange
output.43.name	TEXT		StringRange
output.44.name	TEXT		StringRange
output.45.name	TEXT		StringRange
output.46.name	TEXT		StringRange
output.47.name	TEXT		StringRange
output.48.name	TEXT		StringRange
output.49.name	TEXT		StringRange
output.50.name	TEXT		StringRange
output.51.name	TEXT		StringRange
output.52.name	TEXT		StringRange
output.53.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.54.name	TEXT		StringRange
output.55.name	TEXT		StringRange
output.56.name	TEXT		StringRange
output.57.name	TEXT		StringRange
output.58.name	TEXT		StringRange
output.59.name	TEXT		StringRange
output.60.name	TEXT		StringRange
output.61.name	TEXT		StringRange
output.62.name	TEXT		StringRange
output.63.name	TEXT		StringRange
output.64.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
output.17	CHOICE		EnumRange (OFF)
output.18	CHOICE		EnumRange (OFF)
output.19	CHOICE		EnumRange (OFF)
output.20	CHOICE		EnumRange (OFF)
output.21	CHOICE		EnumRange (OFF)
output.22	CHOICE		EnumRange (OFF)
output.23	CHOICE		EnumRange (OFF)
output.24	CHOICE		EnumRange (OFF)
output.25	CHOICE		EnumRange (OFF)
output.26	CHOICE		EnumRange (OFF)
output.27	CHOICE		EnumRange (OFF)
output.28	CHOICE		EnumRange (OFF)
output.29	CHOICE		EnumRange (OFF)
output.30	CHOICE		EnumRange (OFF)
output.31	CHOICE		EnumRange (OFF)
output.32	CHOICE		EnumRange (OFF)
output.33	CHOICE		EnumRange (OFF)
output.34	CHOICE		EnumRange (OFF)
output.35	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.36	CHOICE		EnumRange (OFF)
output.37	CHOICE		EnumRange (OFF)
output.38	CHOICE		EnumRange (OFF)
output.39	CHOICE		EnumRange (OFF)
output.40	CHOICE		EnumRange (OFF)
output.41	CHOICE		EnumRange (OFF)
output.42	CHOICE		EnumRange (OFF)
output.43	CHOICE		EnumRange (OFF)
output.44	CHOICE		EnumRange (OFF)
output.45	CHOICE		EnumRange (OFF)
output.46	CHOICE		EnumRange (OFF)
output.47	CHOICE		EnumRange (OFF)
output.48	CHOICE		EnumRange (OFF)
output.49	CHOICE		EnumRange (OFF)
output.50	CHOICE		EnumRange (OFF)
output.51	CHOICE		EnumRange (OFF)
output.52	CHOICE		EnumRange (OFF)
output.53	CHOICE		EnumRange (OFF)
output.54	CHOICE		EnumRange (OFF)
output.55	CHOICE		EnumRange (OFF)
output.56	CHOICE		EnumRange (OFF)
output.57	CHOICE		EnumRange (OFF)
output.58	CHOICE		EnumRange (OFF)
output.59	CHOICE		EnumRange (OFF)
output.60	CHOICE		EnumRange (OFF)
output.61	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.62	CHOICE		EnumRange (OFF)
output.63	CHOICE		EnumRange (OFF)
output.64	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.17	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.18	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.19	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.20	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.21	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.22	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.23	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.24	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.25	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.26	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.27	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.28	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.29	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.30	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.31	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.32	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.33	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.34	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.35	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.36	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.37	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.38	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.39	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.40	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.41	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.42	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.43	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.44	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.45	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.46	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.47	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.48	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.49	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.50	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.51	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.52	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.53	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.54	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.55	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.56	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.57	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.58	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.59	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.60	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.61	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.62	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.63	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.64	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
output.01.locking	INTEGER		IntegerRange (0 .. 1)
output.02.locking	INTEGER		IntegerRange (0 .. 1)
output.03.locking	INTEGER		IntegerRange (0 .. 1)
output.04.locking	INTEGER		IntegerRange (0 .. 1)
output.05.locking	INTEGER		IntegerRange (0 .. 1)
output.06.locking	INTEGER		IntegerRange (0 .. 1)
output.07.locking	INTEGER		IntegerRange (0 .. 1)
output.08.locking	INTEGER		IntegerRange (0 .. 1)
output.09.locking	INTEGER		IntegerRange (0 .. 1)
output.10.locking	INTEGER		IntegerRange (0 .. 1)
output.11.locking	INTEGER		IntegerRange (0 .. 1)
output.12.locking	INTEGER		IntegerRange (0 .. 1)
output.13.locking	INTEGER		IntegerRange (0 .. 1)
output.14.locking	INTEGER		IntegerRange (0 .. 1)
output.15.locking	INTEGER		IntegerRange (0 .. 1)
output.16.locking	INTEGER		IntegerRange (0 .. 1)
output.17.locking	INTEGER		IntegerRange (0 .. 1)
output.18.locking	INTEGER		IntegerRange (0 .. 1)
output.19.locking	INTEGER		IntegerRange (0 .. 1)
output.20.locking	INTEGER		IntegerRange (0 .. 1)
output.21.locking	INTEGER		IntegerRange (0 .. 1)
output.22.locking	INTEGER		IntegerRange (0 .. 1)
output.23.locking	INTEGER		IntegerRange (0 .. 1)
output.24.locking	INTEGER		IntegerRange (0 .. 1)
output.25.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.26.locking	INTEGER		IntegerRange (0 .. 1)
output.27.locking	INTEGER		IntegerRange (0 .. 1)
output.28.locking	INTEGER		IntegerRange (0 .. 1)
output.29.locking	INTEGER		IntegerRange (0 .. 1)
output.30.locking	INTEGER		IntegerRange (0 .. 1)
output.31.locking	INTEGER		IntegerRange (0 .. 1)
output.32.locking	INTEGER		IntegerRange (0 .. 1)
output.33.locking	INTEGER		IntegerRange (0 .. 1)
output.34.locking	INTEGER		IntegerRange (0 .. 1)
output.35.locking	INTEGER		IntegerRange (0 .. 1)
output.36.locking	INTEGER		IntegerRange (0 .. 1)
output.37.locking	INTEGER		IntegerRange (0 .. 1)
output.38.locking	INTEGER		IntegerRange (0 .. 1)
output.39.locking	INTEGER		IntegerRange (0 .. 1)
output.40.locking	INTEGER		IntegerRange (0 .. 1)
output.41.locking	INTEGER		IntegerRange (0 .. 1)
output.42.locking	INTEGER		IntegerRange (0 .. 1)
output.43.locking	INTEGER		IntegerRange (0 .. 1)
output.44.locking	INTEGER		IntegerRange (0 .. 1)
output.45.locking	INTEGER		IntegerRange (0 .. 1)
output.46.locking	INTEGER		IntegerRange (0 .. 1)
output.47.locking	INTEGER		IntegerRange (0 .. 1)
output.48.locking	INTEGER		IntegerRange (0 .. 1)
output.49.locking	INTEGER		IntegerRange (0 .. 1)
output.50.locking	INTEGER		IntegerRange (0 .. 1)
output.51.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.52.locking	INTEGER		IntegerRange (0 .. 1)
output.53.locking	INTEGER		IntegerRange (0 .. 1)
output.54.locking	INTEGER		IntegerRange (0 .. 1)
output.55.locking	INTEGER		IntegerRange (0 .. 1)
output.56.locking	INTEGER		IntegerRange (0 .. 1)
output.57.locking	INTEGER		IntegerRange (0 .. 1)
output.58.locking	INTEGER		IntegerRange (0 .. 1)
output.59.locking	INTEGER		IntegerRange (0 .. 1)
output.60.locking	INTEGER		IntegerRange (0 .. 1)
output.61.locking	INTEGER		IntegerRange (0 .. 1)
output.62.locking	INTEGER		IntegerRange (0 .. 1)
output.63.locking	INTEGER		IntegerRange (0 .. 1)
output.64.locking	INTEGER		IntegerRange (0 .. 1)
input.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.01.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.02.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.29.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan PSU2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU1)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Backup Active)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Monitoring)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Monitoring)
faults.101	ALARM	R/O	AlarmFlagRange R/O (Input 1 Summary)
faults.102	ALARM	R/O	AlarmFlagRange R/O (Input 2 Summary)
faults.103	ALARM	R/O	AlarmFlagRange R/O (Input 3 Summary)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Input 4 Summary)

name	*type*	*flags*	*range*
faults.105	ALARM	R/O	AlarmFlagRange R/O (Input 5 Summary)
faults.106	ALARM	R/O	AlarmFlagRange R/O (Input 6 Summary)
faults.107	ALARM	R/O	AlarmFlagRange R/O (Input 7 Summary)
faults.108	ALARM	R/O	AlarmFlagRange R/O (Input 8 Summary)
faults.109	ALARM	R/O	AlarmFlagRange R/O (Input 9 Summary)
faults.110	ALARM	R/O	AlarmFlagRange R/O (Input 10 Summary)
faults.111	ALARM	R/O	AlarmFlagRange R/O (Input 11 Summary)
faults.112	ALARM	R/O	AlarmFlagRange R/O (Input 12 Summary)
faults.113	ALARM	R/O	AlarmFlagRange R/O (Input 13 Summary)
faults.114	ALARM	R/O	AlarmFlagRange R/O (Input 14 Summary)
faults.115	ALARM	R/O	AlarmFlagRange R/O (Input 15 Summary)
faults.116	ALARM	R/O	AlarmFlagRange R/O (Input 16 Summary)
faults.117	ALARM	R/O	AlarmFlagRange R/O (Input 17 Summary)
faults.118	ALARM	R/O	AlarmFlagRange R/O (Input 18 Summary)
faults.119	ALARM	R/O	AlarmFlagRange R/O (Input 19 Summary)
faults.120	ALARM	R/O	AlarmFlagRange R/O (Input 20 Summary)

name	*type*	*flags*	*range*
faults.121	ALARM	R/O	AlarmFlagRange R/O (Input 21 Summary)
faults.122	ALARM	R/O	AlarmFlagRange R/O (Input 22 Summary)
faults.123	ALARM	R/O	AlarmFlagRange R/O (Input 23 Summary)
faults.124	ALARM	R/O	AlarmFlagRange R/O (Input 24 Summary)
faults.125	ALARM	R/O	AlarmFlagRange R/O (Input 25 Summary)
faults.126	ALARM	R/O	AlarmFlagRange R/O (Input 26 Summary)
faults.127	ALARM	R/O	AlarmFlagRange R/O (Input 27 Summary)
faults.128	ALARM	R/O	AlarmFlagRange R/O (Input 28 Summary)
faults.129	ALARM	R/O	AlarmFlagRange R/O (Input 29 Summary)
faults.130	ALARM	R/O	AlarmFlagRange R/O (Input 30 Summary)
faults.131	ALARM	R/O	AlarmFlagRange R/O (Input 31 Summary)
faults.132	ALARM	R/O	AlarmFlagRange R/O (Input 32 Summary)
faults.133	ALARM	R/O	AlarmFlagRange R/O (Input 33 Summary)
faults.134	ALARM	R/O	AlarmFlagRange R/O (Input 34 Summary)
faults.135	ALARM	R/O	AlarmFlagRange R/O (Input 35 Summary)
faults.136	ALARM	R/O	AlarmFlagRange R/O (Input 36 Summary)

name	*type*	*flags*	*range*
faults.137	ALARM	R/O	AlarmFlagRange R/O (Input 37 Summary)
faults.138	ALARM	R/O	AlarmFlagRange R/O (Input 38 Summary)
faults.139	ALARM	R/O	AlarmFlagRange R/O (Input 39 Summary)
faults.140	ALARM	R/O	AlarmFlagRange R/O (Input 40 Summary)
faults.141	ALARM	R/O	AlarmFlagRange R/O (Input 41 Summary)
faults.142	ALARM	R/O	AlarmFlagRange R/O (Input 42 Summary)
faults.143	ALARM	R/O	AlarmFlagRange R/O (Input 43 Summary)
faults.144	ALARM	R/O	AlarmFlagRange R/O (Input 44 Summary)
faults.145	ALARM	R/O	AlarmFlagRange R/O (Input 45 Summary)
faults.146	ALARM	R/O	AlarmFlagRange R/O (Input 46 Summary)
faults.147	ALARM	R/O	AlarmFlagRange R/O (Input 47 Summary)
faults.148	ALARM	R/O	AlarmFlagRange R/O (Input 48 Summary)
faults.149	ALARM	R/O	AlarmFlagRange R/O (Input 49 Summary)
faults.150	ALARM	R/O	AlarmFlagRange R/O (Input 50 Summary)
faults.151	ALARM	R/O	AlarmFlagRange R/O (Input 51 Summary)
faults.152	ALARM	R/O	AlarmFlagRange R/O (Input 52 Summary)

name	*type*	*flags*	*range*
faults.153	ALARM	R/O	AlarmFlagRange R/O (Input 53 Summary)
faults.154	ALARM	R/O	AlarmFlagRange R/O (Input 54 Summary)
faults.155	ALARM	R/O	AlarmFlagRange R/O (Input 55 Summary)
faults.156	ALARM	R/O	AlarmFlagRange R/O (Input 56 Summary)
faults.157	ALARM	R/O	AlarmFlagRange R/O (Input 57 Summary)
faults.158	ALARM	R/O	AlarmFlagRange R/O (Input 58 Summary)
faults.159	ALARM	R/O	AlarmFlagRange R/O (Input 59 Summary)
faults.160	ALARM	R/O	AlarmFlagRange R/O (Input 60 Summary)
faults.161	ALARM	R/O	AlarmFlagRange R/O (Input 61 Summary)
faults.162	ALARM	R/O	AlarmFlagRange R/O (Input 62 Summary)
faults.163	ALARM	R/O	AlarmFlagRange R/O (Input 63 Summary)
faults.164	ALARM	R/O	AlarmFlagRange R/O (Input 64 Summary)
faults.201	ALARM	R/O	AlarmFlagRange R/O (Output 1 Summary)
faults.202	ALARM	R/O	AlarmFlagRange R/O (Output 2 Summary)
faults.203	ALARM	R/O	AlarmFlagRange R/O (Output 3 Summary)
faults.204	ALARM	R/O	AlarmFlagRange R/O (Output 4 Summary)

name	*type*	*flags*	*range*
faults.205	ALARM	R/O	AlarmFlagRange R/O (Output 5 Summary)
faults.206	ALARM	R/O	AlarmFlagRange R/O (Output 6 Summary)
faults.207	ALARM	R/O	AlarmFlagRange R/O (Output 7 Summary)
faults.208	ALARM	R/O	AlarmFlagRange R/O (Output 8 Summary)
faults.209	ALARM	R/O	AlarmFlagRange R/O (Output 9 Summary)
faults.210	ALARM	R/O	AlarmFlagRange R/O (Output 10 Summary)
faults.211	ALARM	R/O	AlarmFlagRange R/O (Output 11 Summary)
faults.212	ALARM	R/O	AlarmFlagRange R/O (Output 12 Summary)
faults.213	ALARM	R/O	AlarmFlagRange R/O (Output 13 Summary)
faults.214	ALARM	R/O	AlarmFlagRange R/O (Output 14 Summary)
faults.215	ALARM	R/O	AlarmFlagRange R/O (Output 15 Summary)
faults.216	ALARM	R/O	AlarmFlagRange R/O (Output 16 Summary)
faults.217	ALARM	R/O	AlarmFlagRange R/O (Output 17 Summary)
faults.218	ALARM	R/O	AlarmFlagRange R/O (Output 18 Summary)
faults.219	ALARM	R/O	AlarmFlagRange R/O (Output 19 Summary)
faults.220	ALARM	R/O	AlarmFlagRange R/O (Output 20 Summary)

name	*type*	*flags*	*range*
faults.221	ALARM	R/O	AlarmFlagRange R/O (Output 21 Summary)
faults.222	ALARM	R/O	AlarmFlagRange R/O (Output 22 Summary)
faults.223	ALARM	R/O	AlarmFlagRange R/O (Output 23 Summary)
faults.224	ALARM	R/O	AlarmFlagRange R/O (Output 24 Summary)
faults.225	ALARM	R/O	AlarmFlagRange R/O (Output 25 Summary)
faults.226	ALARM	R/O	AlarmFlagRange R/O (Output 26 Summary)
faults.227	ALARM	R/O	AlarmFlagRange R/O (Output 27 Summary)
faults.228	ALARM	R/O	AlarmFlagRange R/O (Output 28 Summary)
faults.229	ALARM	R/O	AlarmFlagRange R/O (Output 29 Summary)
faults.230	ALARM	R/O	AlarmFlagRange R/O (Output 30 Summary)
faults.231	ALARM	R/O	AlarmFlagRange R/O (Output 31 Summary)
faults.232	ALARM	R/O	AlarmFlagRange R/O (Output 32 Summary)
faults.233	ALARM	R/O	AlarmFlagRange R/O (Output 33 Summary)
faults.234	ALARM	R/O	AlarmFlagRange R/O (Output 34 Summary)
faults.235	ALARM	R/O	AlarmFlagRange R/O (Output 35 Summary)
faults.236	ALARM	R/O	AlarmFlagRange R/O (Output 36 Summary)

name	*type*	*flags*	*range*
faults.237	ALARM	R/O	AlarmFlagRange R/O (Output 37 Summary)
faults.238	ALARM	R/O	AlarmFlagRange R/O (Output 38 Summary)
faults.239	ALARM	R/O	AlarmFlagRange R/O (Output 39 Summary)
faults.240	ALARM	R/O	AlarmFlagRange R/O (Output 40 Summary)
faults.241	ALARM	R/O	AlarmFlagRange R/O (Output 41 Summary)
faults.242	ALARM	R/O	AlarmFlagRange R/O (Output 42 Summary)
faults.243	ALARM	R/O	AlarmFlagRange R/O (Output 43 Summary)
faults.244	ALARM	R/O	AlarmFlagRange R/O (Output 44 Summary)
faults.245	ALARM	R/O	AlarmFlagRange R/O (Output 45 Summary)
faults.246	ALARM	R/O	AlarmFlagRange R/O (Output 46 Summary)
faults.247	ALARM	R/O	AlarmFlagRange R/O (Output 47 Summary)
faults.248	ALARM	R/O	AlarmFlagRange R/O (Output 48 Summary)
faults.249	ALARM	R/O	AlarmFlagRange R/O (Output 49 Summary)
faults.250	ALARM	R/O	AlarmFlagRange R/O (Output 50 Summary)
faults.251	ALARM	R/O	AlarmFlagRange R/O (Output 51 Summary)
faults.252	ALARM	R/O	AlarmFlagRange R/O (Output 52 Summary)

name	*type*	*flags*	*range*
faults.253	ALARM	R/O	AlarmFlagRange R/O (Output 53 Summary)
faults.254	ALARM	R/O	AlarmFlagRange R/O (Output 54 Summary)
faults.255	ALARM	R/O	AlarmFlagRange R/O (Output 55 Summary)
faults.256	ALARM	R/O	AlarmFlagRange R/O (Output 56 Summary)
faults.257	ALARM	R/O	AlarmFlagRange R/O (Output 57 Summary)
faults.258	ALARM	R/O	AlarmFlagRange R/O (Output 58 Summary)
faults.259	ALARM	R/O	AlarmFlagRange R/O (Output 59 Summary)
faults.260	ALARM	R/O	AlarmFlagRange R/O (Output 60 Summary)
faults.261	ALARM	R/O	AlarmFlagRange R/O (Output 61 Summary)
faults.262	ALARM	R/O	AlarmFlagRange R/O (Output 62 Summary)
faults.263	ALARM	R/O	AlarmFlagRange R/O (Output 63 Summary)
faults.264	ALARM	R/O	AlarmFlagRange R/O (Output 64 Summary)

9.206 ETL-NGM100C-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix NGM-100C series.

The driver works with the "C" model (the combining matrix) only. It supports 32x32 and 32x64 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Port switching by name 1-32.
- --- Port switching by name 33-64.
- --- Port switching by number 1-32.
- --- Port switching by number 33-64.
- --- Redundancy settings.
- --- Edit input names.
- --- Edit output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu3Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu4Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu3FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu4FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.version	TEXT	R/O	StringRange R/O
info.swReleaseDate	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.outputs	TEXT	R/O	StringRange R/O
info.matrixMode	CHOICE		EnumRange (COMBINING DISTRIBUTING)
protection.inhibit	BOOLEAN	NOPRESET	BooleanRange (true false)
aliasRead	TEXT		StringRange
output.01.present	CHOICE	R/O	EnumRange R/O (0 1)
output.02.present	CHOICE	R/O	EnumRange R/O (0 1)
output.03.present	CHOICE	R/O	EnumRange R/O (0 1)
output.04.present	CHOICE	R/O	EnumRange R/O (0 1)
output.05.present	CHOICE	R/O	EnumRange R/O (0 1)
output.06.present	CHOICE	R/O	EnumRange R/O (0 1)
output.07.present	CHOICE	R/O	EnumRange R/O (0 1)
output.08.present	CHOICE	R/O	EnumRange R/O (0 1)
output.09.present	CHOICE	R/O	EnumRange R/O (0 1)
output.10.present	CHOICE	R/O	EnumRange R/O (0 1)
output.11.present	CHOICE	R/O	EnumRange R/O (0 1)
output.12.present	CHOICE	R/O	EnumRange R/O (0 1)
output.13.present	CHOICE	R/O	EnumRange R/O (0 1)
output.14.present	CHOICE	R/O	EnumRange R/O (0 1)
output.15.present	CHOICE	R/O	EnumRange R/O (0 1)
output.16.present	CHOICE	R/O	EnumRange R/O (0 1)
output.17.present	CHOICE	R/O	EnumRange R/O (0 1)
output.18.present	CHOICE	R/O	EnumRange R/O (0 1)
output.19.present	CHOICE	R/O	EnumRange R/O (0 1)
output.20.present	CHOICE	R/O	EnumRange R/O (0 1)
output.21.present	CHOICE	R/O	EnumRange R/O (0 1)
output.22.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.23.present	CHOICE	R/O	EnumRange R/O (0 1)
output.24.present	CHOICE	R/O	EnumRange R/O (0 1)
output.25.present	CHOICE	R/O	EnumRange R/O (0 1)
output.26.present	CHOICE	R/O	EnumRange R/O (0 1)
output.27.present	CHOICE	R/O	EnumRange R/O (0 1)
output.28.present	CHOICE	R/O	EnumRange R/O (0 1)
output.29.present	CHOICE	R/O	EnumRange R/O (0 1)
output.30.present	CHOICE	R/O	EnumRange R/O (0 1)
output.31.present	CHOICE	R/O	EnumRange R/O (0 1)
output.32.present	CHOICE	R/O	EnumRange R/O (0 1)
output.33.present	CHOICE	R/O	EnumRange R/O (0 1)
output.34.present	CHOICE	R/O	EnumRange R/O (0 1)
output.35.present	CHOICE	R/O	EnumRange R/O (0 1)
output.36.present	CHOICE	R/O	EnumRange R/O (0 1)
output.37.present	CHOICE	R/O	EnumRange R/O (0 1)
output.38.present	CHOICE	R/O	EnumRange R/O (0 1)
output.39.present	CHOICE	R/O	EnumRange R/O (0 1)
output.40.present	CHOICE	R/O	EnumRange R/O (0 1)
output.41.present	CHOICE	R/O	EnumRange R/O (0 1)
output.42.present	CHOICE	R/O	EnumRange R/O (0 1)
output.43.present	CHOICE	R/O	EnumRange R/O (0 1)
output.44.present	CHOICE	R/O	EnumRange R/O (0 1)
output.45.present	CHOICE	R/O	EnumRange R/O (0 1)
output.46.present	CHOICE	R/O	EnumRange R/O (0 1)
output.47.present	CHOICE	R/O	EnumRange R/O (0 1)
output.48.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.49.present	CHOICE	R/O	EnumRange R/O (0 1)
output.50.present	CHOICE	R/O	EnumRange R/O (0 1)
output.51.present	CHOICE	R/O	EnumRange R/O (0 1)
output.52.present	CHOICE	R/O	EnumRange R/O (0 1)
output.53.present	CHOICE	R/O	EnumRange R/O (0 1)
output.54.present	CHOICE	R/O	EnumRange R/O (0 1)
output.55.present	CHOICE	R/O	EnumRange R/O (0 1)
output.56.present	CHOICE	R/O	EnumRange R/O (0 1)
output.57.present	CHOICE	R/O	EnumRange R/O (0 1)
output.58.present	CHOICE	R/O	EnumRange R/O (0 1)
output.59.present	CHOICE	R/O	EnumRange R/O (0 1)
output.60.present	CHOICE	R/O	EnumRange R/O (0 1)
output.61.present	CHOICE	R/O	EnumRange R/O (0 1)
output.62.present	CHOICE	R/O	EnumRange R/O (0 1)
output.63.present	CHOICE	R/O	EnumRange R/O (0 1)
output.64.present	CHOICE	R/O	EnumRange R/O (0 1)
output.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)
output.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
output.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)
output.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
output.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
output.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)
output.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
output.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)
output.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)

name	*type*	*flags*	*range*
output.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)
output.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
output.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)
output.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.43.name	TEXT		StringRange
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange
input.55.name	TEXT		StringRange
input.56.name	TEXT		StringRange
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
input.01	CHOICE		EnumRange (OFF)
input.02	CHOICE		EnumRange (OFF)
input.03	CHOICE		EnumRange (OFF)
input.04	CHOICE		EnumRange (OFF)
input.05	CHOICE		EnumRange (OFF)
input.06	CHOICE		EnumRange (OFF)
input.07	CHOICE		EnumRange (OFF)
input.08	CHOICE		EnumRange (OFF)
input.09	CHOICE		EnumRange (OFF)
input.10	CHOICE		EnumRange (OFF)
input.11	CHOICE		EnumRange (OFF)
input.12	CHOICE		EnumRange (OFF)
input.13	CHOICE		EnumRange (OFF)
input.14	CHOICE		EnumRange (OFF)
input.15	CHOICE		EnumRange (OFF)
input.16	CHOICE		EnumRange (OFF)
input.17	CHOICE		EnumRange (OFF)
input.18	CHOICE		EnumRange (OFF)
input.19	CHOICE		EnumRange (OFF)
input.20	CHOICE		EnumRange (OFF)
input.21	CHOICE		EnumRange (OFF)
input.22	CHOICE		EnumRange (OFF)
input.23	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
input.24	CHOICE		EnumRange (OFF)
input.25	CHOICE		EnumRange (OFF)
input.26	CHOICE		EnumRange (OFF)
input.27	CHOICE		EnumRange (OFF)
input.28	CHOICE		EnumRange (OFF)
input.29	CHOICE		EnumRange (OFF)
input.30	CHOICE		EnumRange (OFF)
input.31	CHOICE		EnumRange (OFF)
input.32	CHOICE		EnumRange (OFF)
input.33	CHOICE		EnumRange (OFF)
input.34	CHOICE		EnumRange (OFF)
input.35	CHOICE		EnumRange (OFF)
input.36	CHOICE		EnumRange (OFF)
input.37	CHOICE		EnumRange (OFF)
input.38	CHOICE		EnumRange (OFF)
input.39	CHOICE		EnumRange (OFF)
input.40	CHOICE		EnumRange (OFF)
input.41	CHOICE		EnumRange (OFF)
input.42	CHOICE		EnumRange (OFF)
input.43	CHOICE		EnumRange (OFF)
input.44	CHOICE		EnumRange (OFF)
input.45	CHOICE		EnumRange (OFF)
input.46	CHOICE		EnumRange (OFF)
input.47	CHOICE		EnumRange (OFF)
input.48	CHOICE		EnumRange (OFF)
input.49	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
input.50	CHOICE		EnumRange (OFF)
input.51	CHOICE		EnumRange (OFF)
input.52	CHOICE		EnumRange (OFF)
input.53	CHOICE		EnumRange (OFF)
input.54	CHOICE		EnumRange (OFF)
input.55	CHOICE		EnumRange (OFF)
input.56	CHOICE		EnumRange (OFF)
input.57	CHOICE		EnumRange (OFF)
input.58	CHOICE		EnumRange (OFF)
input.59	CHOICE		EnumRange (OFF)
input.60	CHOICE		EnumRange (OFF)
input.61	CHOICE		EnumRange (OFF)
input.62	CHOICE		EnumRange (OFF)
input.63	CHOICE		EnumRange (OFF)
input.64	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.17	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.18	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.19	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.20	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.21	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.22	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.23	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.24	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.25	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.26	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.27	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.28	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.29	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.30	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.31	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.32	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.33	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.34	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.35	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.36	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.37	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.38	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.39	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.40	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.41	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.42	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.43	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.44	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.45	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.46	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.47	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.48	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.49	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.50	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.51	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.52	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.53	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.54	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.55	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.56	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.57	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.58	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.59	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.60	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.61	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.62	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.63	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.64	FLOAT		DoubleRange (-5.0 .. 5.0)
input.01.locking	INTEGER		IntegerRange (0 .. 1)
input.02.locking	INTEGER		IntegerRange (0 .. 1)
input.03.locking	INTEGER		IntegerRange (0 .. 1)
input.04.locking	INTEGER		IntegerRange (0 .. 1)
input.05.locking	INTEGER		IntegerRange (0 .. 1)
input.06.locking	INTEGER		IntegerRange (0 .. 1)
input.07.locking	INTEGER		IntegerRange (0 .. 1)
input.08.locking	INTEGER		IntegerRange (0 .. 1)
input.09.locking	INTEGER		IntegerRange (0 .. 1)
input.10.locking	INTEGER		IntegerRange (0 .. 1)
input.11.locking	INTEGER		IntegerRange (0 .. 1)
input.12.locking	INTEGER		IntegerRange (0 .. 1)
input.13.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
input.14.locking	INTEGER		IntegerRange (0 .. 1)
input.15.locking	INTEGER		IntegerRange (0 .. 1)
input.16.locking	INTEGER		IntegerRange (0 .. 1)
input.17.locking	INTEGER		IntegerRange (0 .. 1)
input.18.locking	INTEGER		IntegerRange (0 .. 1)
input.19.locking	INTEGER		IntegerRange (0 .. 1)
input.20.locking	INTEGER		IntegerRange (0 .. 1)
input.21.locking	INTEGER		IntegerRange (0 .. 1)
input.22.locking	INTEGER		IntegerRange (0 .. 1)
input.23.locking	INTEGER		IntegerRange (0 .. 1)
input.24.locking	INTEGER		IntegerRange (0 .. 1)
input.25.locking	INTEGER		IntegerRange (0 .. 1)
input.26.locking	INTEGER		IntegerRange (0 .. 1)
input.27.locking	INTEGER		IntegerRange (0 .. 1)
input.28.locking	INTEGER		IntegerRange (0 .. 1)
input.29.locking	INTEGER		IntegerRange (0 .. 1)
input.30.locking	INTEGER		IntegerRange (0 .. 1)
input.31.locking	INTEGER		IntegerRange (0 .. 1)
input.32.locking	INTEGER		IntegerRange (0 .. 1)
input.33.locking	INTEGER		IntegerRange (0 .. 1)
input.34.locking	INTEGER		IntegerRange (0 .. 1)
input.35.locking	INTEGER		IntegerRange (0 .. 1)
input.36.locking	INTEGER		IntegerRange (0 .. 1)
input.37.locking	INTEGER		IntegerRange (0 .. 1)
input.38.locking	INTEGER		IntegerRange (0 .. 1)
input.39.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
input.40.locking	INTEGER		IntegerRange (0 .. 1)
input.41.locking	INTEGER		IntegerRange (0 .. 1)
input.42.locking	INTEGER		IntegerRange (0 .. 1)
input.43.locking	INTEGER		IntegerRange (0 .. 1)
input.44.locking	INTEGER		IntegerRange (0 .. 1)
input.45.locking	INTEGER		IntegerRange (0 .. 1)
input.46.locking	INTEGER		IntegerRange (0 .. 1)
input.47.locking	INTEGER		IntegerRange (0 .. 1)
input.48.locking	INTEGER		IntegerRange (0 .. 1)
input.49.locking	INTEGER		IntegerRange (0 .. 1)
input.50.locking	INTEGER		IntegerRange (0 .. 1)
input.51.locking	INTEGER		IntegerRange (0 .. 1)
input.52.locking	INTEGER		IntegerRange (0 .. 1)
input.53.locking	INTEGER		IntegerRange (0 .. 1)
input.54.locking	INTEGER		IntegerRange (0 .. 1)
input.55.locking	INTEGER		IntegerRange (0 .. 1)
input.56.locking	INTEGER		IntegerRange (0 .. 1)
input.57.locking	INTEGER		IntegerRange (0 .. 1)
input.58.locking	INTEGER		IntegerRange (0 .. 1)
input.59.locking	INTEGER		IntegerRange (0 .. 1)
input.60.locking	INTEGER		IntegerRange (0 .. 1)
input.61.locking	INTEGER		IntegerRange (0 .. 1)
input.62.locking	INTEGER		IntegerRange (0 .. 1)
input.63.locking	INTEGER		IntegerRange (0 .. 1)
input.64.locking	INTEGER		IntegerRange (0 .. 1)
input.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan PSU2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Backup Active)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU2)

9.207 ETL-NGM21C-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix NGM-21C series.

The driver works with the "C" model (the combining matrix) only. It supports 32x32 and 32x64 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-32.
- abc --- Port switching by name 33-64.
- 123 --- Port switching by number 1-32.
- 123 --- Port switching by number 33-64.
- --- Redundancy settings.
- --- Edit input names.
- --- Edit output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu3Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu4Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu3FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu4FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.version	TEXT	R/O	StringRange R/O
info.swReleaseDate	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.matrixMode	CHOICE		EnumRange (COMBINING DISTRIBUTING)
protection.inhibit	BOOLEAN	NOPRESET	BooleanRange (true false)
aliasRead	TEXT		StringRange
output.01.present	CHOICE	R/O	EnumRange R/O (0 1)
output.02.present	CHOICE	R/O	EnumRange R/O (0 1)
output.03.present	CHOICE	R/O	EnumRange R/O (0 1)
output.04.present	CHOICE	R/O	EnumRange R/O (0 1)
output.05.present	CHOICE	R/O	EnumRange R/O (0 1)
output.06.present	CHOICE	R/O	EnumRange R/O (0 1)
output.07.present	CHOICE	R/O	EnumRange R/O (0 1)
output.08.present	CHOICE	R/O	EnumRange R/O (0 1)
output.09.present	CHOICE	R/O	EnumRange R/O (0 1)
output.10.present	CHOICE	R/O	EnumRange R/O (0 1)
output.11.present	CHOICE	R/O	EnumRange R/O (0 1)
output.12.present	CHOICE	R/O	EnumRange R/O (0 1)
output.13.present	CHOICE	R/O	EnumRange R/O (0 1)
output.14.present	CHOICE	R/O	EnumRange R/O (0 1)
output.15.present	CHOICE	R/O	EnumRange R/O (0 1)
output.16.present	CHOICE	R/O	EnumRange R/O (0 1)
output.17.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.18.present	CHOICE	R/O	EnumRange R/O (0 1)
output.19.present	CHOICE	R/O	EnumRange R/O (0 1)
output.20.present	CHOICE	R/O	EnumRange R/O (0 1)
output.21.present	CHOICE	R/O	EnumRange R/O (0 1)
output.22.present	CHOICE	R/O	EnumRange R/O (0 1)
output.23.present	CHOICE	R/O	EnumRange R/O (0 1)
output.24.present	CHOICE	R/O	EnumRange R/O (0 1)
output.25.present	CHOICE	R/O	EnumRange R/O (0 1)
output.26.present	CHOICE	R/O	EnumRange R/O (0 1)
output.27.present	CHOICE	R/O	EnumRange R/O (0 1)
output.28.present	CHOICE	R/O	EnumRange R/O (0 1)
output.29.present	CHOICE	R/O	EnumRange R/O (0 1)
output.30.present	CHOICE	R/O	EnumRange R/O (0 1)
output.31.present	CHOICE	R/O	EnumRange R/O (0 1)
output.32.present	CHOICE	R/O	EnumRange R/O (0 1)
output.33.present	CHOICE	R/O	EnumRange R/O (0 1)
output.34.present	CHOICE	R/O	EnumRange R/O (0 1)
output.35.present	CHOICE	R/O	EnumRange R/O (0 1)
output.36.present	CHOICE	R/O	EnumRange R/O (0 1)
output.37.present	CHOICE	R/O	EnumRange R/O (0 1)
output.38.present	CHOICE	R/O	EnumRange R/O (0 1)
output.39.present	CHOICE	R/O	EnumRange R/O (0 1)
output.40.present	CHOICE	R/O	EnumRange R/O (0 1)
output.41.present	CHOICE	R/O	EnumRange R/O (0 1)
output.42.present	CHOICE	R/O	EnumRange R/O (0 1)
output.43.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.44.present	CHOICE	R/O	EnumRange R/O (0 1)
output.45.present	CHOICE	R/O	EnumRange R/O (0 1)
output.46.present	CHOICE	R/O	EnumRange R/O (0 1)
output.47.present	CHOICE	R/O	EnumRange R/O (0 1)
output.48.present	CHOICE	R/O	EnumRange R/O (0 1)
output.49.present	CHOICE	R/O	EnumRange R/O (0 1)
output.50.present	CHOICE	R/O	EnumRange R/O (0 1)
output.51.present	CHOICE	R/O	EnumRange R/O (0 1)
output.52.present	CHOICE	R/O	EnumRange R/O (0 1)
output.53.present	CHOICE	R/O	EnumRange R/O (0 1)
output.54.present	CHOICE	R/O	EnumRange R/O (0 1)
output.55.present	CHOICE	R/O	EnumRange R/O (0 1)
output.56.present	CHOICE	R/O	EnumRange R/O (0 1)
output.57.present	CHOICE	R/O	EnumRange R/O (0 1)
output.58.present	CHOICE	R/O	EnumRange R/O (0 1)
output.59.present	CHOICE	R/O	EnumRange R/O (0 1)
output.60.present	CHOICE	R/O	EnumRange R/O (0 1)
output.61.present	CHOICE	R/O	EnumRange R/O (0 1)
output.62.present	CHOICE	R/O	EnumRange R/O (0 1)
output.63.present	CHOICE	R/O	EnumRange R/O (0 1)
output.64.present	CHOICE	R/O	EnumRange R/O (0 1)
output.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)
output.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
output.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)
output.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
output.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
output.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)
output.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
output.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)

name	*type*	*flags*	*range*
output.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)
output.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)
output.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
output.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)
output.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange
input.43.name	TEXT		StringRange
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange
input.55.name	TEXT		StringRange
input.56.name	TEXT		StringRange
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
input.01	CHOICE		EnumRange (OFF)
input.02	CHOICE		EnumRange (OFF)
input.03	CHOICE		EnumRange (OFF)
input.04	CHOICE		EnumRange (OFF)
input.05	CHOICE		EnumRange (OFF)
input.06	CHOICE		EnumRange (OFF)
input.07	CHOICE		EnumRange (OFF)
input.08	CHOICE		EnumRange (OFF)
input.09	CHOICE		EnumRange (OFF)
input.10	CHOICE		EnumRange (OFF)
input.11	CHOICE		EnumRange (OFF)
input.12	CHOICE		EnumRange (OFF)
input.13	CHOICE		EnumRange (OFF)
input.14	CHOICE		EnumRange (OFF)
input.15	CHOICE		EnumRange (OFF)
input.16	CHOICE		EnumRange (OFF)
input.17	CHOICE		EnumRange (OFF)
input.18	CHOICE		EnumRange (OFF)
input.19	CHOICE		EnumRange (OFF)
input.20	CHOICE		EnumRange (OFF)
input.21	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
input.22	CHOICE		EnumRange (OFF)
input.23	CHOICE		EnumRange (OFF)
input.24	CHOICE		EnumRange (OFF)
input.25	CHOICE		EnumRange (OFF)
input.26	CHOICE		EnumRange (OFF)
input.27	CHOICE		EnumRange (OFF)
input.28	CHOICE		EnumRange (OFF)
input.29	CHOICE		EnumRange (OFF)
input.30	CHOICE		EnumRange (OFF)
input.31	CHOICE		EnumRange (OFF)
input.32	CHOICE		EnumRange (OFF)
input.33	CHOICE		EnumRange (OFF)
input.34	CHOICE		EnumRange (OFF)
input.35	CHOICE		EnumRange (OFF)
input.36	CHOICE		EnumRange (OFF)
input.37	CHOICE		EnumRange (OFF)
input.38	CHOICE		EnumRange (OFF)
input.39	CHOICE		EnumRange (OFF)
input.40	CHOICE		EnumRange (OFF)
input.41	CHOICE		EnumRange (OFF)
input.42	CHOICE		EnumRange (OFF)
input.43	CHOICE		EnumRange (OFF)
input.44	CHOICE		EnumRange (OFF)
input.45	CHOICE		EnumRange (OFF)
input.46	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
input.47	CHOICE		EnumRange (OFF)
input.48	CHOICE		EnumRange (OFF)
input.49	CHOICE		EnumRange (OFF)
input.50	CHOICE		EnumRange (OFF)
input.51	CHOICE		EnumRange (OFF)
input.52	CHOICE		EnumRange (OFF)
input.53	CHOICE		EnumRange (OFF)
input.54	CHOICE		EnumRange (OFF)
input.55	CHOICE		EnumRange (OFF)
input.56	CHOICE		EnumRange (OFF)
input.57	CHOICE		EnumRange (OFF)
input.58	CHOICE		EnumRange (OFF)
input.59	CHOICE		EnumRange (OFF)
input.60	CHOICE		EnumRange (OFF)
input.61	CHOICE		EnumRange (OFF)
input.62	CHOICE		EnumRange (OFF)
input.63	CHOICE		EnumRange (OFF)
input.64	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
input.01.locking	INTEGER		IntegerRange (0 .. 1)
input.02.locking	INTEGER		IntegerRange (0 .. 1)
input.03.locking	INTEGER		IntegerRange (0 .. 1)
input.04.locking	INTEGER		IntegerRange (0 .. 1)
input.05.locking	INTEGER		IntegerRange (0 .. 1)
input.06.locking	INTEGER		IntegerRange (0 .. 1)
input.07.locking	INTEGER		IntegerRange (0 .. 1)
input.08.locking	INTEGER		IntegerRange (0 .. 1)
input.09.locking	INTEGER		IntegerRange (0 .. 1)
input.10.locking	INTEGER		IntegerRange (0 .. 1)
input.11.locking	INTEGER		IntegerRange (0 .. 1)
input.12.locking	INTEGER		IntegerRange (0 .. 1)
input.13.locking	INTEGER		IntegerRange (0 .. 1)
input.14.locking	INTEGER		IntegerRange (0 .. 1)
input.15.locking	INTEGER		IntegerRange (0 .. 1)
input.16.locking	INTEGER		IntegerRange (0 .. 1)
input.17.locking	INTEGER		IntegerRange (0 .. 1)
input.18.locking	INTEGER		IntegerRange (0 .. 1)
input.19.locking	INTEGER		IntegerRange (0 .. 1)
input.20.locking	INTEGER		IntegerRange (0 .. 1)
input.21.locking	INTEGER		IntegerRange (0 .. 1)
input.22.locking	INTEGER		IntegerRange (0 .. 1)
input.23.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
input.24.locking	INTEGER		IntegerRange (0 .. 1)
input.25.locking	INTEGER		IntegerRange (0 .. 1)
input.26.locking	INTEGER		IntegerRange (0 .. 1)
input.27.locking	INTEGER		IntegerRange (0 .. 1)
input.28.locking	INTEGER		IntegerRange (0 .. 1)
input.29.locking	INTEGER		IntegerRange (0 .. 1)
input.30.locking	INTEGER		IntegerRange (0 .. 1)
input.31.locking	INTEGER		IntegerRange (0 .. 1)
input.32.locking	INTEGER		IntegerRange (0 .. 1)
input.33.locking	INTEGER		IntegerRange (0 .. 1)
input.34.locking	INTEGER		IntegerRange (0 .. 1)
input.35.locking	INTEGER		IntegerRange (0 .. 1)
input.36.locking	INTEGER		IntegerRange (0 .. 1)
input.37.locking	INTEGER		IntegerRange (0 .. 1)
input.38.locking	INTEGER		IntegerRange (0 .. 1)
input.39.locking	INTEGER		IntegerRange (0 .. 1)
input.40.locking	INTEGER		IntegerRange (0 .. 1)
input.41.locking	INTEGER		IntegerRange (0 .. 1)
input.42.locking	INTEGER		IntegerRange (0 .. 1)
input.43.locking	INTEGER		IntegerRange (0 .. 1)
input.44.locking	INTEGER		IntegerRange (0 .. 1)
input.45.locking	INTEGER		IntegerRange (0 .. 1)
input.46.locking	INTEGER		IntegerRange (0 .. 1)
input.47.locking	INTEGER		IntegerRange (0 .. 1)
input.48.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
input.49.locking	INTEGER		IntegerRange (0 .. 1)
input.50.locking	INTEGER		IntegerRange (0 .. 1)
input.51.locking	INTEGER		IntegerRange (0 .. 1)
input.52.locking	INTEGER		IntegerRange (0 .. 1)
input.53.locking	INTEGER		IntegerRange (0 .. 1)
input.54.locking	INTEGER		IntegerRange (0 .. 1)
input.55.locking	INTEGER		IntegerRange (0 .. 1)
input.56.locking	INTEGER		IntegerRange (0 .. 1)
input.57.locking	INTEGER		IntegerRange (0 .. 1)
input.58.locking	INTEGER		IntegerRange (0 .. 1)
input.59.locking	INTEGER		IntegerRange (0 .. 1)
input.60.locking	INTEGER		IntegerRange (0 .. 1)
input.61.locking	INTEGER		IntegerRange (0 .. 1)
input.62.locking	INTEGER		IntegerRange (0 .. 1)
input.63.locking	INTEGER		IntegerRange (0 .. 1)
input.64.locking	INTEGER		IntegerRange (0 .. 1)
input.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan PSU2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Backup Active)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU2)

9.208 ETL-NGM30-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix NGM-30 series.

The driver works with the distributive matrix only. It supports 32x32 and 64x32 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching by name 1-32.
- --- Port switching by name 33-64.
- --- Port switching by number 1-32.
- --- Port switching by number 33-64.
- --- Redundancy settings.
- --- Edit input names.
- --- Edit output names.

- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the READ ALIAS NAMES FROM CHASSIS button on the device driver. The reason is that reading these value is only done initially at the server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu3Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu4Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu3FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
info.psu4FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 20000)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.comFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.psuFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.version	TEXT	R/O	StringRange R/O
info.swReleaseDate	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
aliasRead	TEXT		StringRange
protection.inhibit	BOOLEAN	NOPRESET	BooleanRange (true false)
input.01.present	CHOICE	R/O	EnumRange R/O (0 1)
input.02.present	CHOICE	R/O	EnumRange R/O (0 1)
input.03.present	CHOICE	R/O	EnumRange R/O (0 1)
input.04.present	CHOICE	R/O	EnumRange R/O (0 1)
input.05.present	CHOICE	R/O	EnumRange R/O (0 1)
input.06.present	CHOICE	R/O	EnumRange R/O (0 1)
input.07.present	CHOICE	R/O	EnumRange R/O (0 1)
input.08.present	CHOICE	R/O	EnumRange R/O (0 1)
input.09.present	CHOICE	R/O	EnumRange R/O (0 1)
input.10.present	CHOICE	R/O	EnumRange R/O (0 1)
input.11.present	CHOICE	R/O	EnumRange R/O (0 1)
input.12.present	CHOICE	R/O	EnumRange R/O (0 1)
input.13.present	CHOICE	R/O	EnumRange R/O (0 1)
input.14.present	CHOICE	R/O	EnumRange R/O (0 1)
input.15.present	CHOICE	R/O	EnumRange R/O (0 1)
input.16.present	CHOICE	R/O	EnumRange R/O (0 1)
input.17.present	CHOICE	R/O	EnumRange R/O (0 1)
input.18.present	CHOICE	R/O	EnumRange R/O (0 1)
input.19.present	CHOICE	R/O	EnumRange R/O (0 1)
input.20.present	CHOICE	R/O	EnumRange R/O (0 1)
input.21.present	CHOICE	R/O	EnumRange R/O (0 1)
input.22.present	CHOICE	R/O	EnumRange R/O (0 1)
input.23.present	CHOICE	R/O	EnumRange R/O (0 1)
input.24.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.25.present	CHOICE	R/O	EnumRange R/O (0 1)
input.26.present	CHOICE	R/O	EnumRange R/O (0 1)
input.27.present	CHOICE	R/O	EnumRange R/O (0 1)
input.28.present	CHOICE	R/O	EnumRange R/O (0 1)
input.29.present	CHOICE	R/O	EnumRange R/O (0 1)
input.30.present	CHOICE	R/O	EnumRange R/O (0 1)
input.31.present	CHOICE	R/O	EnumRange R/O (0 1)
input.32.present	CHOICE	R/O	EnumRange R/O (0 1)
input.33.present	CHOICE	R/O	EnumRange R/O (0 1)
input.34.present	CHOICE	R/O	EnumRange R/O (0 1)
input.35.present	CHOICE	R/O	EnumRange R/O (0 1)
input.36.present	CHOICE	R/O	EnumRange R/O (0 1)
input.37.present	CHOICE	R/O	EnumRange R/O (0 1)
input.38.present	CHOICE	R/O	EnumRange R/O (0 1)
input.39.present	CHOICE	R/O	EnumRange R/O (0 1)
input.40.present	CHOICE	R/O	EnumRange R/O (0 1)
input.41.present	CHOICE	R/O	EnumRange R/O (0 1)
input.42.present	CHOICE	R/O	EnumRange R/O (0 1)
input.43.present	CHOICE	R/O	EnumRange R/O (0 1)
input.44.present	CHOICE	R/O	EnumRange R/O (0 1)
input.45.present	CHOICE	R/O	EnumRange R/O (0 1)
input.46.present	CHOICE	R/O	EnumRange R/O (0 1)
input.47.present	CHOICE	R/O	EnumRange R/O (0 1)
input.48.present	CHOICE	R/O	EnumRange R/O (0 1)
input.49.present	CHOICE	R/O	EnumRange R/O (0 1)
input.50.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.51.present	CHOICE	R/O	EnumRange R/O (0 1)
input.52.present	CHOICE	R/O	EnumRange R/O (0 1)
input.53.present	CHOICE	R/O	EnumRange R/O (0 1)
input.54.present	CHOICE	R/O	EnumRange R/O (0 1)
input.55.present	CHOICE	R/O	EnumRange R/O (0 1)
input.56.present	CHOICE	R/O	EnumRange R/O (0 1)
input.57.present	CHOICE	R/O	EnumRange R/O (0 1)
input.58.present	CHOICE	R/O	EnumRange R/O (0 1)
input.59.present	CHOICE	R/O	EnumRange R/O (0 1)
input.60.present	CHOICE	R/O	EnumRange R/O (0 1)
input.61.present	CHOICE	R/O	EnumRange R/O (0 1)
input.62.present	CHOICE	R/O	EnumRange R/O (0 1)
input.63.present	CHOICE	R/O	EnumRange R/O (0 1)
input.64.present	CHOICE	R/O	EnumRange R/O (0 1)
input.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)
input.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
input.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
input.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)
input.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
input.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
input.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)
input.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
input.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)
input.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)
input.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)

name	*type*	*flags*	*range*
input.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
input.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)
input.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
output.33.name	TEXT		StringRange
output.34.name	TEXT		StringRange
output.35.name	TEXT		StringRange
output.36.name	TEXT		StringRange
output.37.name	TEXT		StringRange
output.38.name	TEXT		StringRange
output.39.name	TEXT		StringRange
output.40.name	TEXT		StringRange
output.41.name	TEXT		StringRange
output.42.name	TEXT		StringRange
output.43.name	TEXT		StringRange
output.44.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.45.name	TEXT		StringRange
output.46.name	TEXT		StringRange
output.47.name	TEXT		StringRange
output.48.name	TEXT		StringRange
output.49.name	TEXT		StringRange
output.50.name	TEXT		StringRange
output.51.name	TEXT		StringRange
output.52.name	TEXT		StringRange
output.53.name	TEXT		StringRange
output.54.name	TEXT		StringRange
output.55.name	TEXT		StringRange
output.56.name	TEXT		StringRange
output.57.name	TEXT		StringRange
output.58.name	TEXT		StringRange
output.59.name	TEXT		StringRange
output.60.name	TEXT		StringRange
output.61.name	TEXT		StringRange
output.62.name	TEXT		StringRange
output.63.name	TEXT		StringRange
output.64.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
output.17	CHOICE		EnumRange (OFF)
output.18	CHOICE		EnumRange (OFF)
output.19	CHOICE		EnumRange (OFF)
output.20	CHOICE		EnumRange (OFF)
output.21	CHOICE		EnumRange (OFF)
output.22	CHOICE		EnumRange (OFF)
output.23	CHOICE		EnumRange (OFF)
output.24	CHOICE		EnumRange (OFF)
output.25	CHOICE		EnumRange (OFF)
output.26	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.27	CHOICE		EnumRange (OFF)
output.28	CHOICE		EnumRange (OFF)
output.29	CHOICE		EnumRange (OFF)
output.30	CHOICE		EnumRange (OFF)
output.31	CHOICE		EnumRange (OFF)
output.32	CHOICE		EnumRange (OFF)
output.33	CHOICE		EnumRange (OFF)
output.34	CHOICE		EnumRange (OFF)
output.35	CHOICE		EnumRange (OFF)
output.36	CHOICE		EnumRange (OFF)
output.37	CHOICE		EnumRange (OFF)
output.38	CHOICE		EnumRange (OFF)
output.39	CHOICE		EnumRange (OFF)
output.40	CHOICE		EnumRange (OFF)
output.41	CHOICE		EnumRange (OFF)
output.42	CHOICE		EnumRange (OFF)
output.43	CHOICE		EnumRange (OFF)
output.44	CHOICE		EnumRange (OFF)
output.45	CHOICE		EnumRange (OFF)
output.46	CHOICE		EnumRange (OFF)
output.47	CHOICE		EnumRange (OFF)
output.48	CHOICE		EnumRange (OFF)
output.49	CHOICE		EnumRange (OFF)
output.50	CHOICE		EnumRange (OFF)
output.51	CHOICE		EnumRange (OFF)
output.52	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.53	CHOICE		EnumRange (OFF)
output.54	CHOICE		EnumRange (OFF)
output.55	CHOICE		EnumRange (OFF)
output.56	CHOICE		EnumRange (OFF)
output.57	CHOICE		EnumRange (OFF)
output.58	CHOICE		EnumRange (OFF)
output.59	CHOICE		EnumRange (OFF)
output.60	CHOICE		EnumRange (OFF)
output.61	CHOICE		EnumRange (OFF)
output.62	CHOICE		EnumRange (OFF)
output.63	CHOICE		EnumRange (OFF)
output.64	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.17	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.18	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.19	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.20	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.21	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.22	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.23	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.24	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.25	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.26	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.27	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.28	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.29	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.30	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.31	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.32	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.33	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.34	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.35	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.36	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.37	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.38	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.39	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.40	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.41	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.42	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.43	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.44	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.45	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.46	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.47	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.48	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.49	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.50	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.51	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.52	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.53	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.54	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.55	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.56	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.57	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.58	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.59	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.60	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.61	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.62	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.63	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.64	FLOAT		DoubleRange (-5.0 .. 5.0)
output.01.locking	INTEGER		IntegerRange (0 .. 1)
output.02.locking	INTEGER		IntegerRange (0 .. 1)
output.03.locking	INTEGER		IntegerRange (0 .. 1)
output.04.locking	INTEGER		IntegerRange (0 .. 1)
output.05.locking	INTEGER		IntegerRange (0 .. 1)
output.06.locking	INTEGER		IntegerRange (0 .. 1)
output.07.locking	INTEGER		IntegerRange (0 .. 1)
output.08.locking	INTEGER		IntegerRange (0 .. 1)
output.09.locking	INTEGER		IntegerRange (0 .. 1)
output.10.locking	INTEGER		IntegerRange (0 .. 1)
output.11.locking	INTEGER		IntegerRange (0 .. 1)
output.12.locking	INTEGER		IntegerRange (0 .. 1)
output.13.locking	INTEGER		IntegerRange (0 .. 1)
output.14.locking	INTEGER		IntegerRange (0 .. 1)
output.15.locking	INTEGER		IntegerRange (0 .. 1)
output.16.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.17.locking	INTEGER		IntegerRange (0 .. 1)
output.18.locking	INTEGER		IntegerRange (0 .. 1)
output.19.locking	INTEGER		IntegerRange (0 .. 1)
output.20.locking	INTEGER		IntegerRange (0 .. 1)
output.21.locking	INTEGER		IntegerRange (0 .. 1)
output.22.locking	INTEGER		IntegerRange (0 .. 1)
output.23.locking	INTEGER		IntegerRange (0 .. 1)
output.24.locking	INTEGER		IntegerRange (0 .. 1)
output.25.locking	INTEGER		IntegerRange (0 .. 1)
output.26.locking	INTEGER		IntegerRange (0 .. 1)
output.27.locking	INTEGER		IntegerRange (0 .. 1)
output.28.locking	INTEGER		IntegerRange (0 .. 1)
output.29.locking	INTEGER		IntegerRange (0 .. 1)
output.30.locking	INTEGER		IntegerRange (0 .. 1)
output.31.locking	INTEGER		IntegerRange (0 .. 1)
output.32.locking	INTEGER		IntegerRange (0 .. 1)
output.33.locking	INTEGER		IntegerRange (0 .. 1)
output.34.locking	INTEGER		IntegerRange (0 .. 1)
output.35.locking	INTEGER		IntegerRange (0 .. 1)
output.36.locking	INTEGER		IntegerRange (0 .. 1)
output.37.locking	INTEGER		IntegerRange (0 .. 1)
output.38.locking	INTEGER		IntegerRange (0 .. 1)
output.39.locking	INTEGER		IntegerRange (0 .. 1)
output.40.locking	INTEGER		IntegerRange (0 .. 1)
output.41.locking	INTEGER		IntegerRange (0 .. 1)
output.42.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.43.locking	INTEGER		IntegerRange (0 .. 1)
output.44.locking	INTEGER		IntegerRange (0 .. 1)
output.45.locking	INTEGER		IntegerRange (0 .. 1)
output.46.locking	INTEGER		IntegerRange (0 .. 1)
output.47.locking	INTEGER		IntegerRange (0 .. 1)
output.48.locking	INTEGER		IntegerRange (0 .. 1)
output.49.locking	INTEGER		IntegerRange (0 .. 1)
output.50.locking	INTEGER		IntegerRange (0 .. 1)
output.51.locking	INTEGER		IntegerRange (0 .. 1)
output.52.locking	INTEGER		IntegerRange (0 .. 1)
output.53.locking	INTEGER		IntegerRange (0 .. 1)
output.54.locking	INTEGER		IntegerRange (0 .. 1)
output.55.locking	INTEGER		IntegerRange (0 .. 1)
output.56.locking	INTEGER		IntegerRange (0 .. 1)
output.57.locking	INTEGER		IntegerRange (0 .. 1)
output.58.locking	INTEGER		IntegerRange (0 .. 1)
output.59.locking	INTEGER		IntegerRange (0 .. 1)
output.60.locking	INTEGER		IntegerRange (0 .. 1)
output.61.locking	INTEGER		IntegerRange (0 .. 1)
output.62.locking	INTEGER		IntegerRange (0 .. 1)
output.63.locking	INTEGER		IntegerRange (0 .. 1)
output.64.locking	INTEGER		IntegerRange (0 .. 1)
input.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU1)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan PSU2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Backup Active)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp. PSU2)

9.209 ETL-NSN100-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix "Ensign" NSN-100 series.

It supports 32x32 and 32x64 models. The FW installed has to support the BulkUpdate Command

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching by name 1-16.
- --- Port switching by name 17-32.
- --- Gain settings.
- --- Define redundant outputs.
- --- Define slave outputs.
- --- Edit input names.
- --- Edit output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- *readBackDelay* --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- *inhibitId* --- If this parameter is set, it is watched by the device driver and compared to the *inhibitVal*. If they are equals, all card redundancy functions of the driver are inhibited. This may be used in a dual CPU redundancy configuration to avoid that the driver for the standby CPU perfoms redundancy switches a second time.
- *inhibitVal* --- The value to be compared to the value in the message received for the *inhibitId*.

Slave Mode

When Slave mode is activated an output port which is defined as slave port of another output

will be mirrored. All crosspoint settings will be permanently copied to the slave one. Slave mode can be switched ON/OFF. The Slave mode is implemented within the ***sat-nms*** Software.

Redundancy Mode

When Redundancy is defined for an output all crosspoints will be copied if the main/master output fails. For dual chassis The redundancy is implemented within the ***sat-nms*** Software.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.inhibited	TEXT	R/O	StringRange R/O
info.cpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.cpuTemp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1FanSpeed.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu2FanSpeed.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.tick	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.setDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.inhibitId	TEXT	SAVE SETUP	StringRange
config.inhibitVal	TEXT	SAVE SETUP	StringRange
config.sendTwice	CHOICE	SAVE SETUP	EnumRange (OFF ON)
initialized	BOOLEAN	R/O	BooleanRange R/O (true false)
bulk	TEXT	NOPRESET	StringRange
aliasRead	TEXT		StringRange
slaveMode	CHOICE	SAVE	EnumRange (OFF ON)
output.01.present	CHOICE	R/O	EnumRange R/O (0 1)
output.02.present	CHOICE	R/O	EnumRange R/O (0 1)
output.03.present	CHOICE	R/O	EnumRange R/O (0 1)
output.04.present	CHOICE	R/O	EnumRange R/O (0 1)
output.05.present	CHOICE	R/O	EnumRange R/O (0 1)
output.06.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.07.present	CHOICE	R/O	EnumRange R/O (0 1)
output.08.present	CHOICE	R/O	EnumRange R/O (0 1)
output.09.present	CHOICE	R/O	EnumRange R/O (0 1)
output.10.present	CHOICE	R/O	EnumRange R/O (0 1)
output.11.present	CHOICE	R/O	EnumRange R/O (0 1)
output.12.present	CHOICE	R/O	EnumRange R/O (0 1)
output.13.present	CHOICE	R/O	EnumRange R/O (0 1)
output.14.present	CHOICE	R/O	EnumRange R/O (0 1)
output.15.present	CHOICE	R/O	EnumRange R/O (0 1)
output.16.present	CHOICE	R/O	EnumRange R/O (0 1)
output.17.present	CHOICE	R/O	EnumRange R/O (0 1)
output.18.present	CHOICE	R/O	EnumRange R/O (0 1)
output.19.present	CHOICE	R/O	EnumRange R/O (0 1)
output.20.present	CHOICE	R/O	EnumRange R/O (0 1)
output.21.present	CHOICE	R/O	EnumRange R/O (0 1)
output.22.present	CHOICE	R/O	EnumRange R/O (0 1)
output.23.present	CHOICE	R/O	EnumRange R/O (0 1)
output.24.present	CHOICE	R/O	EnumRange R/O (0 1)
output.25.present	CHOICE	R/O	EnumRange R/O (0 1)
output.26.present	CHOICE	R/O	EnumRange R/O (0 1)
output.27.present	CHOICE	R/O	EnumRange R/O (0 1)
output.28.present	CHOICE	R/O	EnumRange R/O (0 1)
output.29.present	CHOICE	R/O	EnumRange R/O (0 1)
output.30.present	CHOICE	R/O	EnumRange R/O (0 1)
output.31.present	CHOICE	R/O	EnumRange R/O (0 1)
output.32.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.33.present	CHOICE	R/O	EnumRange R/O (0 1)
output.34.present	CHOICE	R/O	EnumRange R/O (0 1)
output.35.present	CHOICE	R/O	EnumRange R/O (0 1)
output.36.present	CHOICE	R/O	EnumRange R/O (0 1)
output.37.present	CHOICE	R/O	EnumRange R/O (0 1)
output.38.present	CHOICE	R/O	EnumRange R/O (0 1)
output.39.present	CHOICE	R/O	EnumRange R/O (0 1)
output.40.present	CHOICE	R/O	EnumRange R/O (0 1)
output.41.present	CHOICE	R/O	EnumRange R/O (0 1)
output.42.present	CHOICE	R/O	EnumRange R/O (0 1)
output.43.present	CHOICE	R/O	EnumRange R/O (0 1)
output.44.present	CHOICE	R/O	EnumRange R/O (0 1)
output.45.present	CHOICE	R/O	EnumRange R/O (0 1)
output.46.present	CHOICE	R/O	EnumRange R/O (0 1)
output.47.present	CHOICE	R/O	EnumRange R/O (0 1)
output.48.present	CHOICE	R/O	EnumRange R/O (0 1)
output.49.present	CHOICE	R/O	EnumRange R/O (0 1)
output.50.present	CHOICE	R/O	EnumRange R/O (0 1)
output.51.present	CHOICE	R/O	EnumRange R/O (0 1)
output.52.present	CHOICE	R/O	EnumRange R/O (0 1)
output.53.present	CHOICE	R/O	EnumRange R/O (0 1)
output.54.present	CHOICE	R/O	EnumRange R/O (0 1)
output.55.present	CHOICE	R/O	EnumRange R/O (0 1)
output.56.present	CHOICE	R/O	EnumRange R/O (0 1)
output.57.present	CHOICE	R/O	EnumRange R/O (0 1)
output.58.present	CHOICE	R/O	EnumRange R/O (0 1)
output.59.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.60.present	CHOICE	R/O	EnumRange R/O (0 1)
output.61.present	CHOICE	R/O	EnumRange R/O (0 1)
output.62.present	CHOICE	R/O	EnumRange R/O (0 1)
output.63.present	CHOICE	R/O	EnumRange R/O (0 1)
output.64.present	CHOICE	R/O	EnumRange R/O (0 1)
input.01.present	CHOICE	R/O	EnumRange R/O (0 1)
input.02.present	CHOICE	R/O	EnumRange R/O (0 1)
input.03.present	CHOICE	R/O	EnumRange R/O (0 1)
input.04.present	CHOICE	R/O	EnumRange R/O (0 1)
input.05.present	CHOICE	R/O	EnumRange R/O (0 1)
input.06.present	CHOICE	R/O	EnumRange R/O (0 1)
input.07.present	CHOICE	R/O	EnumRange R/O (0 1)
input.08.present	CHOICE	R/O	EnumRange R/O (0 1)
input.09.present	CHOICE	R/O	EnumRange R/O (0 1)
input.10.present	CHOICE	R/O	EnumRange R/O (0 1)
input.11.present	CHOICE	R/O	EnumRange R/O (0 1)
input.12.present	CHOICE	R/O	EnumRange R/O (0 1)
input.13.present	CHOICE	R/O	EnumRange R/O (0 1)
input.14.present	CHOICE	R/O	EnumRange R/O (0 1)
input.15.present	CHOICE	R/O	EnumRange R/O (0 1)
input.16.present	CHOICE	R/O	EnumRange R/O (0 1)
input.17.present	CHOICE	R/O	EnumRange R/O (0 1)
input.18.present	CHOICE	R/O	EnumRange R/O (0 1)
input.19.present	CHOICE	R/O	EnumRange R/O (0 1)
input.20.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.21.present	CHOICE	R/O	EnumRange R/O (0 1)
input.22.present	CHOICE	R/O	EnumRange R/O (0 1)
input.23.present	CHOICE	R/O	EnumRange R/O (0 1)
input.24.present	CHOICE	R/O	EnumRange R/O (0 1)
input.25.present	CHOICE	R/O	EnumRange R/O (0 1)
input.26.present	CHOICE	R/O	EnumRange R/O (0 1)
input.27.present	CHOICE	R/O	EnumRange R/O (0 1)
input.28.present	CHOICE	R/O	EnumRange R/O (0 1)
input.29.present	CHOICE	R/O	EnumRange R/O (0 1)
input.30.present	CHOICE	R/O	EnumRange R/O (0 1)
input.31.present	CHOICE	R/O	EnumRange R/O (0 1)
input.32.present	CHOICE	R/O	EnumRange R/O (0 1)
input.33.present	CHOICE	R/O	EnumRange R/O (0 1)
input.34.present	CHOICE	R/O	EnumRange R/O (0 1)
input.35.present	CHOICE	R/O	EnumRange R/O (0 1)
input.36.present	CHOICE	R/O	EnumRange R/O (0 1)
input.37.present	CHOICE	R/O	EnumRange R/O (0 1)
input.38.present	CHOICE	R/O	EnumRange R/O (0 1)
input.39.present	CHOICE	R/O	EnumRange R/O (0 1)
input.40.present	CHOICE	R/O	EnumRange R/O (0 1)
input.41.present	CHOICE	R/O	EnumRange R/O (0 1)
input.42.present	CHOICE	R/O	EnumRange R/O (0 1)
input.43.present	CHOICE	R/O	EnumRange R/O (0 1)
input.44.present	CHOICE	R/O	EnumRange R/O (0 1)
input.45.present	CHOICE	R/O	EnumRange R/O (0 1)
input.46.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.47.present	CHOICE	R/O	EnumRange R/O (0 1)
input.48.present	CHOICE	R/O	EnumRange R/O (0 1)
input.49.present	CHOICE	R/O	EnumRange R/O (0 1)
input.50.present	CHOICE	R/O	EnumRange R/O (0 1)
input.51.present	CHOICE	R/O	EnumRange R/O (0 1)
input.52.present	CHOICE	R/O	EnumRange R/O (0 1)
input.53.present	CHOICE	R/O	EnumRange R/O (0 1)
input.54.present	CHOICE	R/O	EnumRange R/O (0 1)
input.55.present	CHOICE	R/O	EnumRange R/O (0 1)
input.56.present	CHOICE	R/O	EnumRange R/O (0 1)
input.57.present	CHOICE	R/O	EnumRange R/O (0 1)
input.58.present	CHOICE	R/O	EnumRange R/O (0 1)
input.59.present	CHOICE	R/O	EnumRange R/O (0 1)
input.60.present	CHOICE	R/O	EnumRange R/O (0 1)
input.61.present	CHOICE	R/O	EnumRange R/O (0 1)
input.62.present	CHOICE	R/O	EnumRange R/O (0 1)
input.63.present	CHOICE	R/O	EnumRange R/O (0 1)
input.64.present	CHOICE	R/O	EnumRange R/O (0 1)
output.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)
output.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
output.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)
output.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
output.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
output.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)
output.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
output.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)

name	*type*	*flags*	*range*
output.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)
output.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)
output.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
output.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)
output.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
output.01.master	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02.master	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03.master	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.05.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.10.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.14.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.19.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32)
output.21.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32)
output.22.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.23.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32)
output.24.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32)
output.25.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32)
output.26.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32)
output.27.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32)
output.28.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32)
output.29.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32)
output.30.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32)
output.31.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32)

name	*type*	*flags*	*range*
output.32.master	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
input.01.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.24.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.51.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.isActiveBackup	BOOLEAN	R/O	BooleanRange R/O (true false)
input.01.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.02.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.03.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.04.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.05.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.06.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.07.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.08.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.09.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.10.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.11.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.12.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.13.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.14.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.15.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.16.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.17.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.18.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.19.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.20.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.21.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.22.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.23.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.24.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.25.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.26.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.27.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.28.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.29.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.30.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.31.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.32.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.33.redundancy	CHOICE	SAVE	EnumRange (NONE 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.34.redundancy	CHOICE	SAVE	EnumRange (NONE 01 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.35.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.36.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.37.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.38.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.39.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.40.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.41.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.42.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.43.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.44.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.45.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.46.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.47.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.48.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.49.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.50.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.51.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.52.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.53.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.54.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.55.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.56.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.57.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.58.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.59.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.60.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.61.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.62.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
input.63.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.64.redundancy	CHOICE	SAVE	EnumRange (NONE 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange
input.43.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange
input.55.name	TEXT		StringRange
input.56.name	TEXT		StringRange
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.32.name	TEXT		StringRange
input.01.switch	TEXT	NOPRESET	StringRange
input.02.switch	TEXT	NOPRESET	StringRange
input.03.switch	TEXT	NOPRESET	StringRange
input.04.switch	TEXT	NOPRESET	StringRange
input.05.switch	TEXT	NOPRESET	StringRange
input.06.switch	TEXT	NOPRESET	StringRange
input.07.switch	TEXT	NOPRESET	StringRange
input.08.switch	TEXT	NOPRESET	StringRange
input.09.switch	TEXT	NOPRESET	StringRange
input.10.switch	TEXT	NOPRESET	StringRange
input.11.switch	TEXT	NOPRESET	StringRange
input.12.switch	TEXT	NOPRESET	StringRange
input.13.switch	TEXT	NOPRESET	StringRange
input.14.switch	TEXT	NOPRESET	StringRange
input.15.switch	TEXT	NOPRESET	StringRange
input.16.switch	TEXT	NOPRESET	StringRange
input.17.switch	TEXT	NOPRESET	StringRange
input.18.switch	TEXT	NOPRESET	StringRange
input.19.switch	TEXT	NOPRESET	StringRange
input.20.switch	TEXT	NOPRESET	StringRange
input.21.switch	TEXT	NOPRESET	StringRange
input.22.switch	TEXT	NOPRESET	StringRange
input.23.switch	TEXT	NOPRESET	StringRange
input.24.switch	TEXT	NOPRESET	StringRange
input.25.switch	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
input.26.switch	TEXT	NOPRESET	StringRange
input.27.switch	TEXT	NOPRESET	StringRange
input.28.switch	TEXT	NOPRESET	StringRange
input.29.switch	TEXT	NOPRESET	StringRange
input.30.switch	TEXT	NOPRESET	StringRange
input.31.switch	TEXT	NOPRESET	StringRange
input.32.switch	TEXT	NOPRESET	StringRange
input.33.switch	TEXT	NOPRESET	StringRange
input.34.switch	TEXT	NOPRESET	StringRange
input.35.switch	TEXT	NOPRESET	StringRange
input.36.switch	TEXT	NOPRESET	StringRange
input.37.switch	TEXT	NOPRESET	StringRange
input.38.switch	TEXT	NOPRESET	StringRange
input.39.switch	TEXT	NOPRESET	StringRange
input.40.switch	TEXT	NOPRESET	StringRange
input.41.switch	TEXT	NOPRESET	StringRange
input.42.switch	TEXT	NOPRESET	StringRange
input.43.switch	TEXT	NOPRESET	StringRange
input.44.switch	TEXT	NOPRESET	StringRange
input.45.switch	TEXT	NOPRESET	StringRange
input.46.switch	TEXT	NOPRESET	StringRange
input.47.switch	TEXT	NOPRESET	StringRange
input.48.switch	TEXT	NOPRESET	StringRange
input.49.switch	TEXT	NOPRESET	StringRange
input.50.switch	TEXT	NOPRESET	StringRange
input.51.switch	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
input.52.switch	TEXT	NOPRESET	StringRange
input.53.switch	TEXT	NOPRESET	StringRange
input.54.switch	TEXT	NOPRESET	StringRange
input.55.switch	TEXT	NOPRESET	StringRange
input.56.switch	TEXT	NOPRESET	StringRange
input.57.switch	TEXT	NOPRESET	StringRange
input.58.switch	TEXT	NOPRESET	StringRange
input.59.switch	TEXT	NOPRESET	StringRange
input.60.switch	TEXT	NOPRESET	StringRange
input.61.switch	TEXT	NOPRESET	StringRange
input.62.switch	TEXT	NOPRESET	StringRange
input.63.switch	TEXT	NOPRESET	StringRange
input.64.switch	TEXT	NOPRESET	StringRange
output.01.switch	TEXT	NOPRESET	StringRange
output.02.switch	TEXT	NOPRESET	StringRange
output.03.switch	TEXT	NOPRESET	StringRange
output.04.switch	TEXT	NOPRESET	StringRange
output.05.switch	TEXT	NOPRESET	StringRange
output.06.switch	TEXT	NOPRESET	StringRange
output.07.switch	TEXT	NOPRESET	StringRange
output.08.switch	TEXT	NOPRESET	StringRange
output.09.switch	TEXT	NOPRESET	StringRange
output.10.switch	TEXT	NOPRESET	StringRange
output.11.switch	TEXT	NOPRESET	StringRange
output.12.switch	TEXT	NOPRESET	StringRange
output.13.switch	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
output.14.switch	TEXT	NOPRESET	StringRange
output.15.switch	TEXT	NOPRESET	StringRange
output.16.switch	TEXT	NOPRESET	StringRange
output.17.switch	TEXT	NOPRESET	StringRange
output.18.switch	TEXT	NOPRESET	StringRange
output.19.switch	TEXT	NOPRESET	StringRange
output.20.switch	TEXT	NOPRESET	StringRange
output.21.switch	TEXT	NOPRESET	StringRange
output.22.switch	TEXT	NOPRESET	StringRange
output.23.switch	TEXT	NOPRESET	StringRange
output.24.switch	TEXT	NOPRESET	StringRange
output.25.switch	TEXT	NOPRESET	StringRange
output.26.switch	TEXT	NOPRESET	StringRange
output.27.switch	TEXT	NOPRESET	StringRange
output.28.switch	TEXT	NOPRESET	StringRange
output.29.switch	TEXT	NOPRESET	StringRange
output.30.switch	TEXT	NOPRESET	StringRange
output.31.switch	TEXT	NOPRESET	StringRange
output.32.switch	TEXT	NOPRESET	StringRange
input.01.extFaultId	TEXT	SAVE	StringRange
input.02.extFaultId	TEXT	SAVE	StringRange
input.03.extFaultId	TEXT	SAVE	StringRange
input.04.extFaultId	TEXT	SAVE	StringRange
input.05.extFaultId	TEXT	SAVE	StringRange
input.06.extFaultId	TEXT	SAVE	StringRange
input.07.extFaultId	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
input.08.extFaultId	TEXT	SAVE	StringRange
input.09.extFaultId	TEXT	SAVE	StringRange
input.10.extFaultId	TEXT	SAVE	StringRange
input.11.extFaultId	TEXT	SAVE	StringRange
input.12.extFaultId	TEXT	SAVE	StringRange
input.13.extFaultId	TEXT	SAVE	StringRange
input.14.extFaultId	TEXT	SAVE	StringRange
input.15.extFaultId	TEXT	SAVE	StringRange
input.16.extFaultId	TEXT	SAVE	StringRange
input.17.extFaultId	TEXT	SAVE	StringRange
input.18.extFaultId	TEXT	SAVE	StringRange
input.19.extFaultId	TEXT	SAVE	StringRange
input.20.extFaultId	TEXT	SAVE	StringRange
input.21.extFaultId	TEXT	SAVE	StringRange
input.22.extFaultId	TEXT	SAVE	StringRange
input.23.extFaultId	TEXT	SAVE	StringRange
input.24.extFaultId	TEXT	SAVE	StringRange
input.25.extFaultId	TEXT	SAVE	StringRange
input.26.extFaultId	TEXT	SAVE	StringRange
input.27.extFaultId	TEXT	SAVE	StringRange
input.28.extFaultId	TEXT	SAVE	StringRange
input.29.extFaultId	TEXT	SAVE	StringRange
input.30.extFaultId	TEXT	SAVE	StringRange
input.31.extFaultId	TEXT	SAVE	StringRange
input.32.extFaultId	TEXT	SAVE	StringRange
input.33.extFaultId	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
input.34.extFaultId	TEXT	SAVE	StringRange
input.35.extFaultId	TEXT	SAVE	StringRange
input.36.extFaultId	TEXT	SAVE	StringRange
input.37.extFaultId	TEXT	SAVE	StringRange
input.38.extFaultId	TEXT	SAVE	StringRange
input.39.extFaultId	TEXT	SAVE	StringRange
input.40.extFaultId	TEXT	SAVE	StringRange
input.41.extFaultId	TEXT	SAVE	StringRange
input.42.extFaultId	TEXT	SAVE	StringRange
input.43.extFaultId	TEXT	SAVE	StringRange
input.44.extFaultId	TEXT	SAVE	StringRange
input.45.extFaultId	TEXT	SAVE	StringRange
input.46.extFaultId	TEXT	SAVE	StringRange
input.47.extFaultId	TEXT	SAVE	StringRange
input.48.extFaultId	TEXT	SAVE	StringRange
input.49.extFaultId	TEXT	SAVE	StringRange
input.50.extFaultId	TEXT	SAVE	StringRange
input.51.extFaultId	TEXT	SAVE	StringRange
input.52.extFaultId	TEXT	SAVE	StringRange
input.53.extFaultId	TEXT	SAVE	StringRange
input.54.extFaultId	TEXT	SAVE	StringRange
input.55.extFaultId	TEXT	SAVE	StringRange
input.56.extFaultId	TEXT	SAVE	StringRange
input.57.extFaultId	TEXT	SAVE	StringRange
input.58.extFaultId	TEXT	SAVE	StringRange
input.59.extFaultId	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
input.60.extFaultId	TEXT	SAVE	StringRange
input.61.extFaultId	TEXT	SAVE	StringRange
input.62.extFaultId	TEXT	SAVE	StringRange
input.63.extFaultId	TEXT	SAVE	StringRange
input.64.extFaultId	TEXT	SAVE	StringRange
output.01.extFaultId	TEXT	SAVE	StringRange
output.02.extFaultId	TEXT	SAVE	StringRange
output.03.extFaultId	TEXT	SAVE	StringRange
output.04.extFaultId	TEXT	SAVE	StringRange
output.05.extFaultId	TEXT	SAVE	StringRange
output.06.extFaultId	TEXT	SAVE	StringRange
output.07.extFaultId	TEXT	SAVE	StringRange
output.08.extFaultId	TEXT	SAVE	StringRange
output.09.extFaultId	TEXT	SAVE	StringRange
output.10.extFaultId	TEXT	SAVE	StringRange
output.11.extFaultId	TEXT	SAVE	StringRange
output.12.extFaultId	TEXT	SAVE	StringRange
output.13.extFaultId	TEXT	SAVE	StringRange
output.14.extFaultId	TEXT	SAVE	StringRange
output.15.extFaultId	TEXT	SAVE	StringRange
output.16.extFaultId	TEXT	SAVE	StringRange
output.17.extFaultId	TEXT	SAVE	StringRange
output.18.extFaultId	TEXT	SAVE	StringRange
output.19.extFaultId	TEXT	SAVE	StringRange
output.20.extFaultId	TEXT	SAVE	StringRange
output.21.extFaultId	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
output.22.extFaultId	TEXT	SAVE	StringRange
output.23.extFaultId	TEXT	SAVE	StringRange
output.24.extFaultId	TEXT	SAVE	StringRange
output.25.extFaultId	TEXT	SAVE	StringRange
output.26.extFaultId	TEXT	SAVE	StringRange
output.27.extFaultId	TEXT	SAVE	StringRange
output.28.extFaultId	TEXT	SAVE	StringRange
output.29.extFaultId	TEXT	SAVE	StringRange
output.30.extFaultId	TEXT	SAVE	StringRange
output.31.extFaultId	TEXT	SAVE	StringRange
output.32.extFaultId	TEXT	SAVE	StringRange
input.01.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.02.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.03.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.04.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.05.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.06.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.07.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.08.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.09.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.10.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.11.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.12.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.13.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.14.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.15.gain	FLOAT		DoubleRange (0.00 .. 5.00)

name	*type*	*flags*	*range*
input.16.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.17.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.18.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.19.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.20.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.21.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.22.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.23.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.24.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.25.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.26.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.27.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.28.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.29.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.30.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.31.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.32.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.33.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.34.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.35.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.36.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.37.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.38.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.39.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.40.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.41.gain	FLOAT		DoubleRange (0.00 .. 5.00)

name	*type*	*flags*	*range*
input.42.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.43.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.44.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.45.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.46.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.47.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.48.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.49.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.50.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.51.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.52.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.53.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.54.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.55.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.56.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.57.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.58.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.59.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.60.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.61.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.62.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.63.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.64.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.01.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.02.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.03.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.04.gain	FLOAT		DoubleRange (0.00 .. 5.00)

name	*type*	*flags*	*range*
output.05.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.06.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.07.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.08.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.09.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.10.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.11.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.12.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.13.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.14.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.15.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.16.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.17.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.18.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.19.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.20.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.21.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.22.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.23.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.24.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.25.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.26.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.27.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.28.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.29.gain	FLOAT		DoubleRange (0.00 .. 5.00)

name	*type*	*flags*	*range*
output.30.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.31.gain	FLOAT		DoubleRange (0.00 .. 5.00)
output.32.gain	FLOAT		DoubleRange (0.00 .. 5.00)
input.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.01.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.02.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.03.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.04.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.05.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.06.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.07.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.08.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.09.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.10.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.11.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.12.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.13.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.14.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.15.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.16.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.17.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.18.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.19.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.20.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.21.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.22.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.23.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.24.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.25.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.26.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.27.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.28.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.29.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.30.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.31.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.32.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.33.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.34.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.35.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.36.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.37.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.38.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.39.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.40.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.41.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.42.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.43.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.44.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.45.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.46.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.47.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.48.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.49.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.50.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.51.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.52.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.53.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.54.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.55.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.56.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.57.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.58.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.59.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.60.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.61.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.62.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
input.63.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.64.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.09.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.14.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.15.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.16.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.17.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.18.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.19.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.20.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.21.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.22.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.23.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.24.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.25.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.26.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.27.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.28.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.29.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.30.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.31.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
output.32.extFault	BOOLEAN	R/O	BooleanRange R/O (true false)
out.01.in.01	CHOICE		EnumRange (OFF ON)
out.01.in.02	CHOICE		EnumRange (OFF ON)
out.01.in.03	CHOICE		EnumRange (OFF ON)
out.01.in.04	CHOICE		EnumRange (OFF ON)
out.01.in.05	CHOICE		EnumRange (OFF ON)
out.01.in.06	CHOICE		EnumRange (OFF ON)
out.01.in.07	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.01.in.08	CHOICE		EnumRange (OFF ON)
out.01.in.09	CHOICE		EnumRange (OFF ON)
out.01.in.10	CHOICE		EnumRange (OFF ON)
out.01.in.11	CHOICE		EnumRange (OFF ON)
out.01.in.12	CHOICE		EnumRange (OFF ON)
out.01.in.13	CHOICE		EnumRange (OFF ON)
out.01.in.14	CHOICE		EnumRange (OFF ON)
out.01.in.15	CHOICE		EnumRange (OFF ON)
out.01.in.16	CHOICE		EnumRange (OFF ON)
out.01.in.17	CHOICE		EnumRange (OFF ON)
out.01.in.18	CHOICE		EnumRange (OFF ON)
out.01.in.19	CHOICE		EnumRange (OFF ON)
out.01.in.20	CHOICE		EnumRange (OFF ON)
out.01.in.21	CHOICE		EnumRange (OFF ON)
out.01.in.22	CHOICE		EnumRange (OFF ON)
out.01.in.23	CHOICE		EnumRange (OFF ON)
out.01.in.24	CHOICE		EnumRange (OFF ON)
out.01.in.25	CHOICE		EnumRange (OFF ON)
out.01.in.26	CHOICE		EnumRange (OFF ON)
out.01.in.27	CHOICE		EnumRange (OFF ON)
out.01.in.28	CHOICE		EnumRange (OFF ON)
out.01.in.29	CHOICE		EnumRange (OFF ON)
out.01.in.30	CHOICE		EnumRange (OFF ON)
out.01.in.31	CHOICE		EnumRange (OFF ON)
out.01.in.32	CHOICE		EnumRange (OFF ON)
out.01.in.33	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.01.in.34	CHOICE		EnumRange (OFF ON)
out.01.in.35	CHOICE		EnumRange (OFF ON)
out.01.in.36	CHOICE		EnumRange (OFF ON)
out.01.in.37	CHOICE		EnumRange (OFF ON)
out.01.in.38	CHOICE		EnumRange (OFF ON)
out.01.in.39	CHOICE		EnumRange (OFF ON)
out.01.in.40	CHOICE		EnumRange (OFF ON)
out.01.in.41	CHOICE		EnumRange (OFF ON)
out.01.in.42	CHOICE		EnumRange (OFF ON)
out.01.in.43	CHOICE		EnumRange (OFF ON)
out.01.in.44	CHOICE		EnumRange (OFF ON)
out.01.in.45	CHOICE		EnumRange (OFF ON)
out.01.in.46	CHOICE		EnumRange (OFF ON)
out.01.in.47	CHOICE		EnumRange (OFF ON)
out.01.in.48	CHOICE		EnumRange (OFF ON)
out.01.in.49	CHOICE		EnumRange (OFF ON)
out.01.in.50	CHOICE		EnumRange (OFF ON)
out.01.in.51	CHOICE		EnumRange (OFF ON)
out.01.in.52	CHOICE		EnumRange (OFF ON)
out.01.in.53	CHOICE		EnumRange (OFF ON)
out.01.in.54	CHOICE		EnumRange (OFF ON)
out.01.in.55	CHOICE		EnumRange (OFF ON)
out.01.in.56	CHOICE		EnumRange (OFF ON)
out.01.in.57	CHOICE		EnumRange (OFF ON)
out.01.in.58	CHOICE		EnumRange (OFF ON)
out.01.in.59	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.01.in.60	CHOICE		EnumRange (OFF ON)
out.01.in.61	CHOICE		EnumRange (OFF ON)
out.01.in.62	CHOICE		EnumRange (OFF ON)
out.01.in.63	CHOICE		EnumRange (OFF ON)
out.01.in.64	CHOICE		EnumRange (OFF ON)
out.02.in.01	CHOICE		EnumRange (OFF ON)
out.02.in.02	CHOICE		EnumRange (OFF ON)
out.02.in.03	CHOICE		EnumRange (OFF ON)
out.02.in.04	CHOICE		EnumRange (OFF ON)
out.02.in.05	CHOICE		EnumRange (OFF ON)
out.02.in.06	CHOICE		EnumRange (OFF ON)
out.02.in.07	CHOICE		EnumRange (OFF ON)
out.02.in.08	CHOICE		EnumRange (OFF ON)
out.02.in.09	CHOICE		EnumRange (OFF ON)
out.02.in.10	CHOICE		EnumRange (OFF ON)
out.02.in.11	CHOICE		EnumRange (OFF ON)
out.02.in.12	CHOICE		EnumRange (OFF ON)
out.02.in.13	CHOICE		EnumRange (OFF ON)
out.02.in.14	CHOICE		EnumRange (OFF ON)
out.02.in.15	CHOICE		EnumRange (OFF ON)
out.02.in.16	CHOICE		EnumRange (OFF ON)
out.02.in.17	CHOICE		EnumRange (OFF ON)
out.02.in.18	CHOICE		EnumRange (OFF ON)
out.02.in.19	CHOICE		EnumRange (OFF ON)
out.02.in.20	CHOICE		EnumRange (OFF ON)
out.02.in.21	CHOICE		EnumRange (OFF ON)
out.02.in.22	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.02.in.23	CHOICE		EnumRange (OFF ON)
out.02.in.24	CHOICE		EnumRange (OFF ON)
out.02.in.25	CHOICE		EnumRange (OFF ON)
out.02.in.26	CHOICE		EnumRange (OFF ON)
out.02.in.27	CHOICE		EnumRange (OFF ON)
out.02.in.28	CHOICE		EnumRange (OFF ON)
out.02.in.29	CHOICE		EnumRange (OFF ON)
out.02.in.30	CHOICE		EnumRange (OFF ON)
out.02.in.31	CHOICE		EnumRange (OFF ON)
out.02.in.32	CHOICE		EnumRange (OFF ON)
out.02.in.33	CHOICE		EnumRange (OFF ON)
out.02.in.34	CHOICE		EnumRange (OFF ON)
out.02.in.35	CHOICE		EnumRange (OFF ON)
out.02.in.36	CHOICE		EnumRange (OFF ON)
out.02.in.37	CHOICE		EnumRange (OFF ON)
out.02.in.38	CHOICE		EnumRange (OFF ON)
out.02.in.39	CHOICE		EnumRange (OFF ON)
out.02.in.40	CHOICE		EnumRange (OFF ON)
out.02.in.41	CHOICE		EnumRange (OFF ON)
out.02.in.42	CHOICE		EnumRange (OFF ON)
out.02.in.43	CHOICE		EnumRange (OFF ON)
out.02.in.44	CHOICE		EnumRange (OFF ON)
out.02.in.45	CHOICE		EnumRange (OFF ON)
out.02.in.46	CHOICE		EnumRange (OFF ON)
out.02.in.47	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.02.in.48	CHOICE		EnumRange (OFF ON)
out.02.in.49	CHOICE		EnumRange (OFF ON)
out.02.in.50	CHOICE		EnumRange (OFF ON)
out.02.in.51	CHOICE		EnumRange (OFF ON)
out.02.in.52	CHOICE		EnumRange (OFF ON)
out.02.in.53	CHOICE		EnumRange (OFF ON)
out.02.in.54	CHOICE		EnumRange (OFF ON)
out.02.in.55	CHOICE		EnumRange (OFF ON)
out.02.in.56	CHOICE		EnumRange (OFF ON)
out.02.in.57	CHOICE		EnumRange (OFF ON)
out.02.in.58	CHOICE		EnumRange (OFF ON)
out.02.in.59	CHOICE		EnumRange (OFF ON)
out.02.in.60	CHOICE		EnumRange (OFF ON)
out.02.in.61	CHOICE		EnumRange (OFF ON)
out.02.in.62	CHOICE		EnumRange (OFF ON)
out.02.in.63	CHOICE		EnumRange (OFF ON)
out.02.in.64	CHOICE		EnumRange (OFF ON)
out.03.in.01	CHOICE		EnumRange (OFF ON)
out.03.in.02	CHOICE		EnumRange (OFF ON)
out.03.in.03	CHOICE		EnumRange (OFF ON)
out.03.in.04	CHOICE		EnumRange (OFF ON)
out.03.in.05	CHOICE		EnumRange (OFF ON)
out.03.in.06	CHOICE		EnumRange (OFF ON)
out.03.in.07	CHOICE		EnumRange (OFF ON)
out.03.in.08	CHOICE		EnumRange (OFF ON)
out.03.in.09	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.03.in.10	CHOICE		EnumRange (OFF ON)
out.03.in.11	CHOICE		EnumRange (OFF ON)
out.03.in.12	CHOICE		EnumRange (OFF ON)
out.03.in.13	CHOICE		EnumRange (OFF ON)
out.03.in.14	CHOICE		EnumRange (OFF ON)
out.03.in.15	CHOICE		EnumRange (OFF ON)
out.03.in.16	CHOICE		EnumRange (OFF ON)
out.03.in.17	CHOICE		EnumRange (OFF ON)
out.03.in.18	CHOICE		EnumRange (OFF ON)
out.03.in.19	CHOICE		EnumRange (OFF ON)
out.03.in.20	CHOICE		EnumRange (OFF ON)
out.03.in.21	CHOICE		EnumRange (OFF ON)
out.03.in.22	CHOICE		EnumRange (OFF ON)
out.03.in.23	CHOICE		EnumRange (OFF ON)
out.03.in.24	CHOICE		EnumRange (OFF ON)
out.03.in.25	CHOICE		EnumRange (OFF ON)
out.03.in.26	CHOICE		EnumRange (OFF ON)
out.03.in.27	CHOICE		EnumRange (OFF ON)
out.03.in.28	CHOICE		EnumRange (OFF ON)
out.03.in.29	CHOICE		EnumRange (OFF ON)
out.03.in.30	CHOICE		EnumRange (OFF ON)
out.03.in.31	CHOICE		EnumRange (OFF ON)
out.03.in.32	CHOICE		EnumRange (OFF ON)
out.03.in.33	CHOICE		EnumRange (OFF ON)
out.03.in.34	CHOICE		EnumRange (OFF ON)
out.03.in.35	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.03.in.36	CHOICE		EnumRange (OFF ON)
out.03.in.37	CHOICE		EnumRange (OFF ON)
out.03.in.38	CHOICE		EnumRange (OFF ON)
out.03.in.39	CHOICE		EnumRange (OFF ON)
out.03.in.40	CHOICE		EnumRange (OFF ON)
out.03.in.41	CHOICE		EnumRange (OFF ON)
out.03.in.42	CHOICE		EnumRange (OFF ON)
out.03.in.43	CHOICE		EnumRange (OFF ON)
out.03.in.44	CHOICE		EnumRange (OFF ON)
out.03.in.45	CHOICE		EnumRange (OFF ON)
out.03.in.46	CHOICE		EnumRange (OFF ON)
out.03.in.47	CHOICE		EnumRange (OFF ON)
out.03.in.48	CHOICE		EnumRange (OFF ON)
out.03.in.49	CHOICE		EnumRange (OFF ON)
out.03.in.50	CHOICE		EnumRange (OFF ON)
out.03.in.51	CHOICE		EnumRange (OFF ON)
out.03.in.52	CHOICE		EnumRange (OFF ON)
out.03.in.53	CHOICE		EnumRange (OFF ON)
out.03.in.54	CHOICE		EnumRange (OFF ON)
out.03.in.55	CHOICE		EnumRange (OFF ON)
out.03.in.56	CHOICE		EnumRange (OFF ON)
out.03.in.57	CHOICE		EnumRange (OFF ON)
out.03.in.58	CHOICE		EnumRange (OFF ON)
out.03.in.59	CHOICE		EnumRange (OFF ON)
out.03.in.60	CHOICE		EnumRange (OFF ON)
out.03.in.61	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.03.in.62	CHOICE		EnumRange (OFF ON)
out.03.in.63	CHOICE		EnumRange (OFF ON)
out.03.in.64	CHOICE		EnumRange (OFF ON)
out.04.in.01	CHOICE		EnumRange (OFF ON)
out.04.in.02	CHOICE		EnumRange (OFF ON)
out.04.in.03	CHOICE		EnumRange (OFF ON)
out.04.in.04	CHOICE		EnumRange (OFF ON)
out.04.in.05	CHOICE		EnumRange (OFF ON)
out.04.in.06	CHOICE		EnumRange (OFF ON)
out.04.in.07	CHOICE		EnumRange (OFF ON)
out.04.in.08	CHOICE		EnumRange (OFF ON)
out.04.in.09	CHOICE		EnumRange (OFF ON)
out.04.in.10	CHOICE		EnumRange (OFF ON)
out.04.in.11	CHOICE		EnumRange (OFF ON)
out.04.in.12	CHOICE		EnumRange (OFF ON)
out.04.in.13	CHOICE		EnumRange (OFF ON)
out.04.in.14	CHOICE		EnumRange (OFF ON)
out.04.in.15	CHOICE		EnumRange (OFF ON)
out.04.in.16	CHOICE		EnumRange (OFF ON)
out.04.in.17	CHOICE		EnumRange (OFF ON)
out.04.in.18	CHOICE		EnumRange (OFF ON)
out.04.in.19	CHOICE		EnumRange (OFF ON)
out.04.in.20	CHOICE		EnumRange (OFF ON)
out.04.in.21	CHOICE		EnumRange (OFF ON)
out.04.in.22	CHOICE		EnumRange (OFF ON)
out.04.in.23	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.04.in.24	CHOICE		EnumRange (OFF ON)
out.04.in.25	CHOICE		EnumRange (OFF ON)
out.04.in.26	CHOICE		EnumRange (OFF ON)
out.04.in.27	CHOICE		EnumRange (OFF ON)
out.04.in.28	CHOICE		EnumRange (OFF ON)
out.04.in.29	CHOICE		EnumRange (OFF ON)
out.04.in.30	CHOICE		EnumRange (OFF ON)
out.04.in.31	CHOICE		EnumRange (OFF ON)
out.04.in.32	CHOICE		EnumRange (OFF ON)
out.04.in.33	CHOICE		EnumRange (OFF ON)
out.04.in.34	CHOICE		EnumRange (OFF ON)
out.04.in.35	CHOICE		EnumRange (OFF ON)
out.04.in.36	CHOICE		EnumRange (OFF ON)
out.04.in.37	CHOICE		EnumRange (OFF ON)
out.04.in.38	CHOICE		EnumRange (OFF ON)
out.04.in.39	CHOICE		EnumRange (OFF ON)
out.04.in.40	CHOICE		EnumRange (OFF ON)
out.04.in.41	CHOICE		EnumRange (OFF ON)
out.04.in.42	CHOICE		EnumRange (OFF ON)
out.04.in.43	CHOICE		EnumRange (OFF ON)
out.04.in.44	CHOICE		EnumRange (OFF ON)
out.04.in.45	CHOICE		EnumRange (OFF ON)
out.04.in.46	CHOICE		EnumRange (OFF ON)
out.04.in.47	CHOICE		EnumRange (OFF ON)
out.04.in.48	CHOICE		EnumRange (OFF ON)
out.04.in.49	CHOICE		EnumRange (OFF ON)
out.04.in.50	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.04.in.51	CHOICE		EnumRange (OFF ON)
out.04.in.52	CHOICE		EnumRange (OFF ON)
out.04.in.53	CHOICE		EnumRange (OFF ON)
out.04.in.54	CHOICE		EnumRange (OFF ON)
out.04.in.55	CHOICE		EnumRange (OFF ON)
out.04.in.56	CHOICE		EnumRange (OFF ON)
out.04.in.57	CHOICE		EnumRange (OFF ON)
out.04.in.58	CHOICE		EnumRange (OFF ON)
out.04.in.59	CHOICE		EnumRange (OFF ON)
out.04.in.60	CHOICE		EnumRange (OFF ON)
out.04.in.61	CHOICE		EnumRange (OFF ON)
out.04.in.62	CHOICE		EnumRange (OFF ON)
out.04.in.63	CHOICE		EnumRange (OFF ON)
out.04.in.64	CHOICE		EnumRange (OFF ON)
out.05.in.01	CHOICE		EnumRange (OFF ON)
out.05.in.02	CHOICE		EnumRange (OFF ON)
out.05.in.03	CHOICE		EnumRange (OFF ON)
out.05.in.04	CHOICE		EnumRange (OFF ON)
out.05.in.05	CHOICE		EnumRange (OFF ON)
out.05.in.06	CHOICE		EnumRange (OFF ON)
out.05.in.07	CHOICE		EnumRange (OFF ON)
out.05.in.08	CHOICE		EnumRange (OFF ON)
out.05.in.09	CHOICE		EnumRange (OFF ON)
out.05.in.10	CHOICE		EnumRange (OFF ON)
out.05.in.11	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.05.in.12	CHOICE		EnumRange (OFF ON)
out.05.in.13	CHOICE		EnumRange (OFF ON)
out.05.in.14	CHOICE		EnumRange (OFF ON)
out.05.in.15	CHOICE		EnumRange (OFF ON)
out.05.in.16	CHOICE		EnumRange (OFF ON)
out.05.in.17	CHOICE		EnumRange (OFF ON)
out.05.in.18	CHOICE		EnumRange (OFF ON)
out.05.in.19	CHOICE		EnumRange (OFF ON)
out.05.in.20	CHOICE		EnumRange (OFF ON)
out.05.in.21	CHOICE		EnumRange (OFF ON)
out.05.in.22	CHOICE		EnumRange (OFF ON)
out.05.in.23	CHOICE		EnumRange (OFF ON)
out.05.in.24	CHOICE		EnumRange (OFF ON)
out.05.in.25	CHOICE		EnumRange (OFF ON)
out.05.in.26	CHOICE		EnumRange (OFF ON)
out.05.in.27	CHOICE		EnumRange (OFF ON)
out.05.in.28	CHOICE		EnumRange (OFF ON)
out.05.in.29	CHOICE		EnumRange (OFF ON)
out.05.in.30	CHOICE		EnumRange (OFF ON)
out.05.in.31	CHOICE		EnumRange (OFF ON)
out.05.in.32	CHOICE		EnumRange (OFF ON)
out.05.in.33	CHOICE		EnumRange (OFF ON)
out.05.in.34	CHOICE		EnumRange (OFF ON)
out.05.in.35	CHOICE		EnumRange (OFF ON)
out.05.in.36	CHOICE		EnumRange (OFF ON)
out.05.in.37	CHOICE		EnumRange (OFF ON)
out.05.in.38	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.05.in.39	CHOICE		EnumRange (OFF ON)
out.05.in.40	CHOICE		EnumRange (OFF ON)
out.05.in.41	CHOICE		EnumRange (OFF ON)
out.05.in.42	CHOICE		EnumRange (OFF ON)
out.05.in.43	CHOICE		EnumRange (OFF ON)
out.05.in.44	CHOICE		EnumRange (OFF ON)
out.05.in.45	CHOICE		EnumRange (OFF ON)
out.05.in.46	CHOICE		EnumRange (OFF ON)
out.05.in.47	CHOICE		EnumRange (OFF ON)
out.05.in.48	CHOICE		EnumRange (OFF ON)
out.05.in.49	CHOICE		EnumRange (OFF ON)
out.05.in.50	CHOICE		EnumRange (OFF ON)
out.05.in.51	CHOICE		EnumRange (OFF ON)
out.05.in.52	CHOICE		EnumRange (OFF ON)
out.05.in.53	CHOICE		EnumRange (OFF ON)
out.05.in.54	CHOICE		EnumRange (OFF ON)
out.05.in.55	CHOICE		EnumRange (OFF ON)
out.05.in.56	CHOICE		EnumRange (OFF ON)
out.05.in.57	CHOICE		EnumRange (OFF ON)
out.05.in.58	CHOICE		EnumRange (OFF ON)
out.05.in.59	CHOICE		EnumRange (OFF ON)
out.05.in.60	CHOICE		EnumRange (OFF ON)
out.05.in.61	CHOICE		EnumRange (OFF ON)
out.05.in.62	CHOICE		EnumRange (OFF ON)
out.05.in.63	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.05.in.64	CHOICE		EnumRange (OFF ON)
out.06.in.01	CHOICE		EnumRange (OFF ON)
out.06.in.02	CHOICE		EnumRange (OFF ON)
out.06.in.03	CHOICE		EnumRange (OFF ON)
out.06.in.04	CHOICE		EnumRange (OFF ON)
out.06.in.05	CHOICE		EnumRange (OFF ON)
out.06.in.06	CHOICE		EnumRange (OFF ON)
out.06.in.07	CHOICE		EnumRange (OFF ON)
out.06.in.08	CHOICE		EnumRange (OFF ON)
out.06.in.09	CHOICE		EnumRange (OFF ON)
out.06.in.10	CHOICE		EnumRange (OFF ON)
out.06.in.11	CHOICE		EnumRange (OFF ON)
out.06.in.12	CHOICE		EnumRange (OFF ON)
out.06.in.13	CHOICE		EnumRange (OFF ON)
out.06.in.14	CHOICE		EnumRange (OFF ON)
out.06.in.15	CHOICE		EnumRange (OFF ON)
out.06.in.16	CHOICE		EnumRange (OFF ON)
out.06.in.17	CHOICE		EnumRange (OFF ON)
out.06.in.18	CHOICE		EnumRange (OFF ON)
out.06.in.19	CHOICE		EnumRange (OFF ON)
out.06.in.20	CHOICE		EnumRange (OFF ON)
out.06.in.21	CHOICE		EnumRange (OFF ON)
out.06.in.22	CHOICE		EnumRange (OFF ON)
out.06.in.23	CHOICE		EnumRange (OFF ON)
out.06.in.24	CHOICE		EnumRange (OFF ON)
out.06.in.25	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.06.in.26	CHOICE		EnumRange (OFF ON)
out.06.in.27	CHOICE		EnumRange (OFF ON)
out.06.in.28	CHOICE		EnumRange (OFF ON)
out.06.in.29	CHOICE		EnumRange (OFF ON)
out.06.in.30	CHOICE		EnumRange (OFF ON)
out.06.in.31	CHOICE		EnumRange (OFF ON)
out.06.in.32	CHOICE		EnumRange (OFF ON)
out.06.in.33	CHOICE		EnumRange (OFF ON)
out.06.in.34	CHOICE		EnumRange (OFF ON)
out.06.in.35	CHOICE		EnumRange (OFF ON)
out.06.in.36	CHOICE		EnumRange (OFF ON)
out.06.in.37	CHOICE		EnumRange (OFF ON)
out.06.in.38	CHOICE		EnumRange (OFF ON)
out.06.in.39	CHOICE		EnumRange (OFF ON)
out.06.in.40	CHOICE		EnumRange (OFF ON)
out.06.in.41	CHOICE		EnumRange (OFF ON)
out.06.in.42	CHOICE		EnumRange (OFF ON)
out.06.in.43	CHOICE		EnumRange (OFF ON)
out.06.in.44	CHOICE		EnumRange (OFF ON)
out.06.in.45	CHOICE		EnumRange (OFF ON)
out.06.in.46	CHOICE		EnumRange (OFF ON)
out.06.in.47	CHOICE		EnumRange (OFF ON)
out.06.in.48	CHOICE		EnumRange (OFF ON)
out.06.in.49	CHOICE		EnumRange (OFF ON)
out.06.in.50	CHOICE		EnumRange (OFF ON)
out.06.in.51	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.06.in.52	CHOICE		EnumRange (OFF ON)
out.06.in.53	CHOICE		EnumRange (OFF ON)
out.06.in.54	CHOICE		EnumRange (OFF ON)
out.06.in.55	CHOICE		EnumRange (OFF ON)
out.06.in.56	CHOICE		EnumRange (OFF ON)
out.06.in.57	CHOICE		EnumRange (OFF ON)
out.06.in.58	CHOICE		EnumRange (OFF ON)
out.06.in.59	CHOICE		EnumRange (OFF ON)
out.06.in.60	CHOICE		EnumRange (OFF ON)
out.06.in.61	CHOICE		EnumRange (OFF ON)
out.06.in.62	CHOICE		EnumRange (OFF ON)
out.06.in.63	CHOICE		EnumRange (OFF ON)
out.06.in.64	CHOICE		EnumRange (OFF ON)
out.07.in.01	CHOICE		EnumRange (OFF ON)
out.07.in.02	CHOICE		EnumRange (OFF ON)
out.07.in.03	CHOICE		EnumRange (OFF ON)
out.07.in.04	CHOICE		EnumRange (OFF ON)
out.07.in.05	CHOICE		EnumRange (OFF ON)
out.07.in.06	CHOICE		EnumRange (OFF ON)
out.07.in.07	CHOICE		EnumRange (OFF ON)
out.07.in.08	CHOICE		EnumRange (OFF ON)
out.07.in.09	CHOICE		EnumRange (OFF ON)
out.07.in.10	CHOICE		EnumRange (OFF ON)
out.07.in.11	CHOICE		EnumRange (OFF ON)
out.07.in.12	CHOICE		EnumRange (OFF ON)
out.07.in.13	CHOICE		EnumRange (OFF ON)
out.07.in.14	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.07.in.15	CHOICE		EnumRange (OFF ON)
out.07.in.16	CHOICE		EnumRange (OFF ON)
out.07.in.17	CHOICE		EnumRange (OFF ON)
out.07.in.18	CHOICE		EnumRange (OFF ON)
out.07.in.19	CHOICE		EnumRange (OFF ON)
out.07.in.20	CHOICE		EnumRange (OFF ON)
out.07.in.21	CHOICE		EnumRange (OFF ON)
out.07.in.22	CHOICE		EnumRange (OFF ON)
out.07.in.23	CHOICE		EnumRange (OFF ON)
out.07.in.24	CHOICE		EnumRange (OFF ON)
out.07.in.25	CHOICE		EnumRange (OFF ON)
out.07.in.26	CHOICE		EnumRange (OFF ON)
out.07.in.27	CHOICE		EnumRange (OFF ON)
out.07.in.28	CHOICE		EnumRange (OFF ON)
out.07.in.29	CHOICE		EnumRange (OFF ON)
out.07.in.30	CHOICE		EnumRange (OFF ON)
out.07.in.31	CHOICE		EnumRange (OFF ON)
out.07.in.32	CHOICE		EnumRange (OFF ON)
out.07.in.33	CHOICE		EnumRange (OFF ON)
out.07.in.34	CHOICE		EnumRange (OFF ON)
out.07.in.35	CHOICE		EnumRange (OFF ON)
out.07.in.36	CHOICE		EnumRange (OFF ON)
out.07.in.37	CHOICE		EnumRange (OFF ON)
out.07.in.38	CHOICE		EnumRange (OFF ON)
out.07.in.39	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.07.in.40	CHOICE		EnumRange (OFF ON)
out.07.in.41	CHOICE		EnumRange (OFF ON)
out.07.in.42	CHOICE		EnumRange (OFF ON)
out.07.in.43	CHOICE		EnumRange (OFF ON)
out.07.in.44	CHOICE		EnumRange (OFF ON)
out.07.in.45	CHOICE		EnumRange (OFF ON)
out.07.in.46	CHOICE		EnumRange (OFF ON)
out.07.in.47	CHOICE		EnumRange (OFF ON)
out.07.in.48	CHOICE		EnumRange (OFF ON)
out.07.in.49	CHOICE		EnumRange (OFF ON)
out.07.in.50	CHOICE		EnumRange (OFF ON)
out.07.in.51	CHOICE		EnumRange (OFF ON)
out.07.in.52	CHOICE		EnumRange (OFF ON)
out.07.in.53	CHOICE		EnumRange (OFF ON)
out.07.in.54	CHOICE		EnumRange (OFF ON)
out.07.in.55	CHOICE		EnumRange (OFF ON)
out.07.in.56	CHOICE		EnumRange (OFF ON)
out.07.in.57	CHOICE		EnumRange (OFF ON)
out.07.in.58	CHOICE		EnumRange (OFF ON)
out.07.in.59	CHOICE		EnumRange (OFF ON)
out.07.in.60	CHOICE		EnumRange (OFF ON)
out.07.in.61	CHOICE		EnumRange (OFF ON)
out.07.in.62	CHOICE		EnumRange (OFF ON)
out.07.in.63	CHOICE		EnumRange (OFF ON)
out.07.in.64	CHOICE		EnumRange (OFF ON)
out.08.in.01	CHOICE		EnumRange (OFF ON)
out.08.in.02	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.08.in.03	CHOICE		EnumRange (OFF ON)
out.08.in.04	CHOICE		EnumRange (OFF ON)
out.08.in.05	CHOICE		EnumRange (OFF ON)
out.08.in.06	CHOICE		EnumRange (OFF ON)
out.08.in.07	CHOICE		EnumRange (OFF ON)
out.08.in.08	CHOICE		EnumRange (OFF ON)
out.08.in.09	CHOICE		EnumRange (OFF ON)
out.08.in.10	CHOICE		EnumRange (OFF ON)
out.08.in.11	CHOICE		EnumRange (OFF ON)
out.08.in.12	CHOICE		EnumRange (OFF ON)
out.08.in.13	CHOICE		EnumRange (OFF ON)
out.08.in.14	CHOICE		EnumRange (OFF ON)
out.08.in.15	CHOICE		EnumRange (OFF ON)
out.08.in.16	CHOICE		EnumRange (OFF ON)
out.08.in.17	CHOICE		EnumRange (OFF ON)
out.08.in.18	CHOICE		EnumRange (OFF ON)
out.08.in.19	CHOICE		EnumRange (OFF ON)
out.08.in.20	CHOICE		EnumRange (OFF ON)
out.08.in.21	CHOICE		EnumRange (OFF ON)
out.08.in.22	CHOICE		EnumRange (OFF ON)
out.08.in.23	CHOICE		EnumRange (OFF ON)
out.08.in.24	CHOICE		EnumRange (OFF ON)
out.08.in.25	CHOICE		EnumRange (OFF ON)
out.08.in.26	CHOICE		EnumRange (OFF ON)
out.08.in.27	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.08.in.28	CHOICE		EnumRange (OFF ON)
out.08.in.29	CHOICE		EnumRange (OFF ON)
out.08.in.30	CHOICE		EnumRange (OFF ON)
out.08.in.31	CHOICE		EnumRange (OFF ON)
out.08.in.32	CHOICE		EnumRange (OFF ON)
out.08.in.33	CHOICE		EnumRange (OFF ON)
out.08.in.34	CHOICE		EnumRange (OFF ON)
out.08.in.35	CHOICE		EnumRange (OFF ON)
out.08.in.36	CHOICE		EnumRange (OFF ON)
out.08.in.37	CHOICE		EnumRange (OFF ON)
out.08.in.38	CHOICE		EnumRange (OFF ON)
out.08.in.39	CHOICE		EnumRange (OFF ON)
out.08.in.40	CHOICE		EnumRange (OFF ON)
out.08.in.41	CHOICE		EnumRange (OFF ON)
out.08.in.42	CHOICE		EnumRange (OFF ON)
out.08.in.43	CHOICE		EnumRange (OFF ON)
out.08.in.44	CHOICE		EnumRange (OFF ON)
out.08.in.45	CHOICE		EnumRange (OFF ON)
out.08.in.46	CHOICE		EnumRange (OFF ON)
out.08.in.47	CHOICE		EnumRange (OFF ON)
out.08.in.48	CHOICE		EnumRange (OFF ON)
out.08.in.49	CHOICE		EnumRange (OFF ON)
out.08.in.50	CHOICE		EnumRange (OFF ON)
out.08.in.51	CHOICE		EnumRange (OFF ON)
out.08.in.52	CHOICE		EnumRange (OFF ON)
out.08.in.53	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.08.in.54	CHOICE		EnumRange (OFF ON)
out.08.in.55	CHOICE		EnumRange (OFF ON)
out.08.in.56	CHOICE		EnumRange (OFF ON)
out.08.in.57	CHOICE		EnumRange (OFF ON)
out.08.in.58	CHOICE		EnumRange (OFF ON)
out.08.in.59	CHOICE		EnumRange (OFF ON)
out.08.in.60	CHOICE		EnumRange (OFF ON)
out.08.in.61	CHOICE		EnumRange (OFF ON)
out.08.in.62	CHOICE		EnumRange (OFF ON)
out.08.in.63	CHOICE		EnumRange (OFF ON)
out.08.in.64	CHOICE		EnumRange (OFF ON)
out.09.in.01	CHOICE		EnumRange (OFF ON)
out.09.in.02	CHOICE		EnumRange (OFF ON)
out.09.in.03	CHOICE		EnumRange (OFF ON)
out.09.in.04	CHOICE		EnumRange (OFF ON)
out.09.in.05	CHOICE		EnumRange (OFF ON)
out.09.in.06	CHOICE		EnumRange (OFF ON)
out.09.in.07	CHOICE		EnumRange (OFF ON)
out.09.in.08	CHOICE		EnumRange (OFF ON)
out.09.in.09	CHOICE		EnumRange (OFF ON)
out.09.in.10	CHOICE		EnumRange (OFF ON)
out.09.in.11	CHOICE		EnumRange (OFF ON)
out.09.in.12	CHOICE		EnumRange (OFF ON)
out.09.in.13	CHOICE		EnumRange (OFF ON)
out.09.in.14	CHOICE		EnumRange (OFF ON)
out.09.in.15	CHOICE		EnumRange (OFF ON)
out.09.in.16	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.09.in.17	CHOICE		EnumRange (OFF ON)
out.09.in.18	CHOICE		EnumRange (OFF ON)
out.09.in.19	CHOICE		EnumRange (OFF ON)
out.09.in.20	CHOICE		EnumRange (OFF ON)
out.09.in.21	CHOICE		EnumRange (OFF ON)
out.09.in.22	CHOICE		EnumRange (OFF ON)
out.09.in.23	CHOICE		EnumRange (OFF ON)
out.09.in.24	CHOICE		EnumRange (OFF ON)
out.09.in.25	CHOICE		EnumRange (OFF ON)
out.09.in.26	CHOICE		EnumRange (OFF ON)
out.09.in.27	CHOICE		EnumRange (OFF ON)
out.09.in.28	CHOICE		EnumRange (OFF ON)
out.09.in.29	CHOICE		EnumRange (OFF ON)
out.09.in.30	CHOICE		EnumRange (OFF ON)
out.09.in.31	CHOICE		EnumRange (OFF ON)
out.09.in.32	CHOICE		EnumRange (OFF ON)
out.09.in.33	CHOICE		EnumRange (OFF ON)
out.09.in.34	CHOICE		EnumRange (OFF ON)
out.09.in.35	CHOICE		EnumRange (OFF ON)
out.09.in.36	CHOICE		EnumRange (OFF ON)
out.09.in.37	CHOICE		EnumRange (OFF ON)
out.09.in.38	CHOICE		EnumRange (OFF ON)
out.09.in.39	CHOICE		EnumRange (OFF ON)
out.09.in.40	CHOICE		EnumRange (OFF ON)
out.09.in.41	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.09.in.42	CHOICE		EnumRange (OFF ON)
out.09.in.43	CHOICE		EnumRange (OFF ON)
out.09.in.44	CHOICE		EnumRange (OFF ON)
out.09.in.45	CHOICE		EnumRange (OFF ON)
out.09.in.46	CHOICE		EnumRange (OFF ON)
out.09.in.47	CHOICE		EnumRange (OFF ON)
out.09.in.48	CHOICE		EnumRange (OFF ON)
out.09.in.49	CHOICE		EnumRange (OFF ON)
out.09.in.50	CHOICE		EnumRange (OFF ON)
out.09.in.51	CHOICE		EnumRange (OFF ON)
out.09.in.52	CHOICE		EnumRange (OFF ON)
out.09.in.53	CHOICE		EnumRange (OFF ON)
out.09.in.54	CHOICE		EnumRange (OFF ON)
out.09.in.55	CHOICE		EnumRange (OFF ON)
out.09.in.56	CHOICE		EnumRange (OFF ON)
out.09.in.57	CHOICE		EnumRange (OFF ON)
out.09.in.58	CHOICE		EnumRange (OFF ON)
out.09.in.59	CHOICE		EnumRange (OFF ON)
out.09.in.60	CHOICE		EnumRange (OFF ON)
out.09.in.61	CHOICE		EnumRange (OFF ON)
out.09.in.62	CHOICE		EnumRange (OFF ON)
out.09.in.63	CHOICE		EnumRange (OFF ON)
out.09.in.64	CHOICE		EnumRange (OFF ON)
out.10.in.01	CHOICE		EnumRange (OFF ON)
out.10.in.02	CHOICE		EnumRange (OFF ON)
out.10.in.03	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.10.in.04	CHOICE		EnumRange (OFF ON)
out.10.in.05	CHOICE		EnumRange (OFF ON)
out.10.in.06	CHOICE		EnumRange (OFF ON)
out.10.in.07	CHOICE		EnumRange (OFF ON)
out.10.in.08	CHOICE		EnumRange (OFF ON)
out.10.in.09	CHOICE		EnumRange (OFF ON)
out.10.in.10	CHOICE		EnumRange (OFF ON)
out.10.in.11	CHOICE		EnumRange (OFF ON)
out.10.in.12	CHOICE		EnumRange (OFF ON)
out.10.in.13	CHOICE		EnumRange (OFF ON)
out.10.in.14	CHOICE		EnumRange (OFF ON)
out.10.in.15	CHOICE		EnumRange (OFF ON)
out.10.in.16	CHOICE		EnumRange (OFF ON)
out.10.in.17	CHOICE		EnumRange (OFF ON)
out.10.in.18	CHOICE		EnumRange (OFF ON)
out.10.in.19	CHOICE		EnumRange (OFF ON)
out.10.in.20	CHOICE		EnumRange (OFF ON)
out.10.in.21	CHOICE		EnumRange (OFF ON)
out.10.in.22	CHOICE		EnumRange (OFF ON)
out.10.in.23	CHOICE		EnumRange (OFF ON)
out.10.in.24	CHOICE		EnumRange (OFF ON)
out.10.in.25	CHOICE		EnumRange (OFF ON)
out.10.in.26	CHOICE		EnumRange (OFF ON)
out.10.in.27	CHOICE		EnumRange (OFF ON)
out.10.in.28	CHOICE		EnumRange (OFF ON)
out.10.in.29	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.10.in.30	CHOICE		EnumRange (OFF ON)
out.10.in.31	CHOICE		EnumRange (OFF ON)
out.10.in.32	CHOICE		EnumRange (OFF ON)
out.10.in.33	CHOICE		EnumRange (OFF ON)
out.10.in.34	CHOICE		EnumRange (OFF ON)
out.10.in.35	CHOICE		EnumRange (OFF ON)
out.10.in.36	CHOICE		EnumRange (OFF ON)
out.10.in.37	CHOICE		EnumRange (OFF ON)
out.10.in.38	CHOICE		EnumRange (OFF ON)
out.10.in.39	CHOICE		EnumRange (OFF ON)
out.10.in.40	CHOICE		EnumRange (OFF ON)
out.10.in.41	CHOICE		EnumRange (OFF ON)
out.10.in.42	CHOICE		EnumRange (OFF ON)
out.10.in.43	CHOICE		EnumRange (OFF ON)
out.10.in.44	CHOICE		EnumRange (OFF ON)
out.10.in.45	CHOICE		EnumRange (OFF ON)
out.10.in.46	CHOICE		EnumRange (OFF ON)
out.10.in.47	CHOICE		EnumRange (OFF ON)
out.10.in.48	CHOICE		EnumRange (OFF ON)
out.10.in.49	CHOICE		EnumRange (OFF ON)
out.10.in.50	CHOICE		EnumRange (OFF ON)
out.10.in.51	CHOICE		EnumRange (OFF ON)
out.10.in.52	CHOICE		EnumRange (OFF ON)
out.10.in.53	CHOICE		EnumRange (OFF ON)
out.10.in.54	CHOICE		EnumRange (OFF ON)
out.10.in.55	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.10.in.56	CHOICE		EnumRange (OFF ON)
out.10.in.57	CHOICE		EnumRange (OFF ON)
out.10.in.58	CHOICE		EnumRange (OFF ON)
out.10.in.59	CHOICE		EnumRange (OFF ON)
out.10.in.60	CHOICE		EnumRange (OFF ON)
out.10.in.61	CHOICE		EnumRange (OFF ON)
out.10.in.62	CHOICE		EnumRange (OFF ON)
out.10.in.63	CHOICE		EnumRange (OFF ON)
out.10.in.64	CHOICE		EnumRange (OFF ON)
out.11.in.01	CHOICE		EnumRange (OFF ON)
out.11.in.02	CHOICE		EnumRange (OFF ON)
out.11.in.03	CHOICE		EnumRange (OFF ON)
out.11.in.04	CHOICE		EnumRange (OFF ON)
out.11.in.05	CHOICE		EnumRange (OFF ON)
out.11.in.06	CHOICE		EnumRange (OFF ON)
out.11.in.07	CHOICE		EnumRange (OFF ON)
out.11.in.08	CHOICE		EnumRange (OFF ON)
out.11.in.09	CHOICE		EnumRange (OFF ON)
out.11.in.10	CHOICE		EnumRange (OFF ON)
out.11.in.11	CHOICE		EnumRange (OFF ON)
out.11.in.12	CHOICE		EnumRange (OFF ON)
out.11.in.13	CHOICE		EnumRange (OFF ON)
out.11.in.14	CHOICE		EnumRange (OFF ON)
out.11.in.15	CHOICE		EnumRange (OFF ON)
out.11.in.16	CHOICE		EnumRange (OFF ON)
out.11.in.17	CHOICE		EnumRange (OFF ON)
out.11.in.18	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.11.in.19	CHOICE		EnumRange (OFF ON)
out.11.in.20	CHOICE		EnumRange (OFF ON)
out.11.in.21	CHOICE		EnumRange (OFF ON)
out.11.in.22	CHOICE		EnumRange (OFF ON)
out.11.in.23	CHOICE		EnumRange (OFF ON)
out.11.in.24	CHOICE		EnumRange (OFF ON)
out.11.in.25	CHOICE		EnumRange (OFF ON)
out.11.in.26	CHOICE		EnumRange (OFF ON)
out.11.in.27	CHOICE		EnumRange (OFF ON)
out.11.in.28	CHOICE		EnumRange (OFF ON)
out.11.in.29	CHOICE		EnumRange (OFF ON)
out.11.in.30	CHOICE		EnumRange (OFF ON)
out.11.in.31	CHOICE		EnumRange (OFF ON)
out.11.in.32	CHOICE		EnumRange (OFF ON)
out.11.in.33	CHOICE		EnumRange (OFF ON)
out.11.in.34	CHOICE		EnumRange (OFF ON)
out.11.in.35	CHOICE		EnumRange (OFF ON)
out.11.in.36	CHOICE		EnumRange (OFF ON)
out.11.in.37	CHOICE		EnumRange (OFF ON)
out.11.in.38	CHOICE		EnumRange (OFF ON)
out.11.in.39	CHOICE		EnumRange (OFF ON)
out.11.in.40	CHOICE		EnumRange (OFF ON)
out.11.in.41	CHOICE		EnumRange (OFF ON)
out.11.in.42	CHOICE		EnumRange (OFF ON)
out.11.in.43	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.11.in.44	CHOICE		EnumRange (OFF ON)
out.11.in.45	CHOICE		EnumRange (OFF ON)
out.11.in.46	CHOICE		EnumRange (OFF ON)
out.11.in.47	CHOICE		EnumRange (OFF ON)
out.11.in.48	CHOICE		EnumRange (OFF ON)
out.11.in.49	CHOICE		EnumRange (OFF ON)
out.11.in.50	CHOICE		EnumRange (OFF ON)
out.11.in.51	CHOICE		EnumRange (OFF ON)
out.11.in.52	CHOICE		EnumRange (OFF ON)
out.11.in.53	CHOICE		EnumRange (OFF ON)
out.11.in.54	CHOICE		EnumRange (OFF ON)
out.11.in.55	CHOICE		EnumRange (OFF ON)
out.11.in.56	CHOICE		EnumRange (OFF ON)
out.11.in.57	CHOICE		EnumRange (OFF ON)
out.11.in.58	CHOICE		EnumRange (OFF ON)
out.11.in.59	CHOICE		EnumRange (OFF ON)
out.11.in.60	CHOICE		EnumRange (OFF ON)
out.11.in.61	CHOICE		EnumRange (OFF ON)
out.11.in.62	CHOICE		EnumRange (OFF ON)
out.11.in.63	CHOICE		EnumRange (OFF ON)
out.11.in.64	CHOICE		EnumRange (OFF ON)
out.12.in.01	CHOICE		EnumRange (OFF ON)
out.12.in.02	CHOICE		EnumRange (OFF ON)
out.12.in.03	CHOICE		EnumRange (OFF ON)
out.12.in.04	CHOICE		EnumRange (OFF ON)
out.12.in.05	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.12.in.06	CHOICE		EnumRange (OFF ON)
out.12.in.07	CHOICE		EnumRange (OFF ON)
out.12.in.08	CHOICE		EnumRange (OFF ON)
out.12.in.09	CHOICE		EnumRange (OFF ON)
out.12.in.10	CHOICE		EnumRange (OFF ON)
out.12.in.11	CHOICE		EnumRange (OFF ON)
out.12.in.12	CHOICE		EnumRange (OFF ON)
out.12.in.13	CHOICE		EnumRange (OFF ON)
out.12.in.14	CHOICE		EnumRange (OFF ON)
out.12.in.15	CHOICE		EnumRange (OFF ON)
out.12.in.16	CHOICE		EnumRange (OFF ON)
out.12.in.17	CHOICE		EnumRange (OFF ON)
out.12.in.18	CHOICE		EnumRange (OFF ON)
out.12.in.19	CHOICE		EnumRange (OFF ON)
out.12.in.20	CHOICE		EnumRange (OFF ON)
out.12.in.21	CHOICE		EnumRange (OFF ON)
out.12.in.22	CHOICE		EnumRange (OFF ON)
out.12.in.23	CHOICE		EnumRange (OFF ON)
out.12.in.24	CHOICE		EnumRange (OFF ON)
out.12.in.25	CHOICE		EnumRange (OFF ON)
out.12.in.26	CHOICE		EnumRange (OFF ON)
out.12.in.27	CHOICE		EnumRange (OFF ON)
out.12.in.28	CHOICE		EnumRange (OFF ON)
out.12.in.29	CHOICE		EnumRange (OFF ON)
out.12.in.30	CHOICE		EnumRange (OFF ON)
out.12.in.31	CHOICE		EnumRange (OFF ON)
out.12.in.32	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.12.in.33	CHOICE		EnumRange (OFF ON)
out.12.in.34	CHOICE		EnumRange (OFF ON)
out.12.in.35	CHOICE		EnumRange (OFF ON)
out.12.in.36	CHOICE		EnumRange (OFF ON)
out.12.in.37	CHOICE		EnumRange (OFF ON)
out.12.in.38	CHOICE		EnumRange (OFF ON)
out.12.in.39	CHOICE		EnumRange (OFF ON)
out.12.in.40	CHOICE		EnumRange (OFF ON)
out.12.in.41	CHOICE		EnumRange (OFF ON)
out.12.in.42	CHOICE		EnumRange (OFF ON)
out.12.in.43	CHOICE		EnumRange (OFF ON)
out.12.in.44	CHOICE		EnumRange (OFF ON)
out.12.in.45	CHOICE		EnumRange (OFF ON)
out.12.in.46	CHOICE		EnumRange (OFF ON)
out.12.in.47	CHOICE		EnumRange (OFF ON)
out.12.in.48	CHOICE		EnumRange (OFF ON)
out.12.in.49	CHOICE		EnumRange (OFF ON)
out.12.in.50	CHOICE		EnumRange (OFF ON)
out.12.in.51	CHOICE		EnumRange (OFF ON)
out.12.in.52	CHOICE		EnumRange (OFF ON)
out.12.in.53	CHOICE		EnumRange (OFF ON)
out.12.in.54	CHOICE		EnumRange (OFF ON)
out.12.in.55	CHOICE		EnumRange (OFF ON)
out.12.in.56	CHOICE		EnumRange (OFF ON)
out.12.in.57	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.12.in.58	CHOICE		EnumRange (OFF ON)
out.12.in.59	CHOICE		EnumRange (OFF ON)
out.12.in.60	CHOICE		EnumRange (OFF ON)
out.12.in.61	CHOICE		EnumRange (OFF ON)
out.12.in.62	CHOICE		EnumRange (OFF ON)
out.12.in.63	CHOICE		EnumRange (OFF ON)
out.12.in.64	CHOICE		EnumRange (OFF ON)
out.13.in.01	CHOICE		EnumRange (OFF ON)
out.13.in.02	CHOICE		EnumRange (OFF ON)
out.13.in.03	CHOICE		EnumRange (OFF ON)
out.13.in.04	CHOICE		EnumRange (OFF ON)
out.13.in.05	CHOICE		EnumRange (OFF ON)
out.13.in.06	CHOICE		EnumRange (OFF ON)
out.13.in.07	CHOICE		EnumRange (OFF ON)
out.13.in.08	CHOICE		EnumRange (OFF ON)
out.13.in.09	CHOICE		EnumRange (OFF ON)
out.13.in.10	CHOICE		EnumRange (OFF ON)
out.13.in.11	CHOICE		EnumRange (OFF ON)
out.13.in.12	CHOICE		EnumRange (OFF ON)
out.13.in.13	CHOICE		EnumRange (OFF ON)
out.13.in.14	CHOICE		EnumRange (OFF ON)
out.13.in.15	CHOICE		EnumRange (OFF ON)
out.13.in.16	CHOICE		EnumRange (OFF ON)
out.13.in.17	CHOICE		EnumRange (OFF ON)
out.13.in.18	CHOICE		EnumRange (OFF ON)
out.13.in.19	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.13.in.20	CHOICE		EnumRange (OFF ON)
out.13.in.21	CHOICE		EnumRange (OFF ON)
out.13.in.22	CHOICE		EnumRange (OFF ON)
out.13.in.23	CHOICE		EnumRange (OFF ON)
out.13.in.24	CHOICE		EnumRange (OFF ON)
out.13.in.25	CHOICE		EnumRange (OFF ON)
out.13.in.26	CHOICE		EnumRange (OFF ON)
out.13.in.27	CHOICE		EnumRange (OFF ON)
out.13.in.28	CHOICE		EnumRange (OFF ON)
out.13.in.29	CHOICE		EnumRange (OFF ON)
out.13.in.30	CHOICE		EnumRange (OFF ON)
out.13.in.31	CHOICE		EnumRange (OFF ON)
out.13.in.32	CHOICE		EnumRange (OFF ON)
out.13.in.33	CHOICE		EnumRange (OFF ON)
out.13.in.34	CHOICE		EnumRange (OFF ON)
out.13.in.35	CHOICE		EnumRange (OFF ON)
out.13.in.36	CHOICE		EnumRange (OFF ON)
out.13.in.37	CHOICE		EnumRange (OFF ON)
out.13.in.38	CHOICE		EnumRange (OFF ON)
out.13.in.39	CHOICE		EnumRange (OFF ON)
out.13.in.40	CHOICE		EnumRange (OFF ON)
out.13.in.41	CHOICE		EnumRange (OFF ON)
out.13.in.42	CHOICE		EnumRange (OFF ON)
out.13.in.43	CHOICE		EnumRange (OFF ON)
out.13.in.44	CHOICE		EnumRange (OFF ON)
out.13.in.45	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.13.in.46	CHOICE		EnumRange (OFF ON)
out.13.in.47	CHOICE		EnumRange (OFF ON)
out.13.in.48	CHOICE		EnumRange (OFF ON)
out.13.in.49	CHOICE		EnumRange (OFF ON)
out.13.in.50	CHOICE		EnumRange (OFF ON)
out.13.in.51	CHOICE		EnumRange (OFF ON)
out.13.in.52	CHOICE		EnumRange (OFF ON)
out.13.in.53	CHOICE		EnumRange (OFF ON)
out.13.in.54	CHOICE		EnumRange (OFF ON)
out.13.in.55	CHOICE		EnumRange (OFF ON)
out.13.in.56	CHOICE		EnumRange (OFF ON)
out.13.in.57	CHOICE		EnumRange (OFF ON)
out.13.in.58	CHOICE		EnumRange (OFF ON)
out.13.in.59	CHOICE		EnumRange (OFF ON)
out.13.in.60	CHOICE		EnumRange (OFF ON)
out.13.in.61	CHOICE		EnumRange (OFF ON)
out.13.in.62	CHOICE		EnumRange (OFF ON)
out.13.in.63	CHOICE		EnumRange (OFF ON)
out.13.in.64	CHOICE		EnumRange (OFF ON)
out.14.in.01	CHOICE		EnumRange (OFF ON)
out.14.in.02	CHOICE		EnumRange (OFF ON)
out.14.in.03	CHOICE		EnumRange (OFF ON)
out.14.in.04	CHOICE		EnumRange (OFF ON)
out.14.in.05	CHOICE		EnumRange (OFF ON)
out.14.in.06	CHOICE		EnumRange (OFF ON)
out.14.in.07	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.14.in.08	CHOICE		EnumRange (OFF ON)
out.14.in.09	CHOICE		EnumRange (OFF ON)
out.14.in.10	CHOICE		EnumRange (OFF ON)
out.14.in.11	CHOICE		EnumRange (OFF ON)
out.14.in.12	CHOICE		EnumRange (OFF ON)
out.14.in.13	CHOICE		EnumRange (OFF ON)
out.14.in.14	CHOICE		EnumRange (OFF ON)
out.14.in.15	CHOICE		EnumRange (OFF ON)
out.14.in.16	CHOICE		EnumRange (OFF ON)
out.14.in.17	CHOICE		EnumRange (OFF ON)
out.14.in.18	CHOICE		EnumRange (OFF ON)
out.14.in.19	CHOICE		EnumRange (OFF ON)
out.14.in.20	CHOICE		EnumRange (OFF ON)
out.14.in.21	CHOICE		EnumRange (OFF ON)
out.14.in.22	CHOICE		EnumRange (OFF ON)
out.14.in.23	CHOICE		EnumRange (OFF ON)
out.14.in.24	CHOICE		EnumRange (OFF ON)
out.14.in.25	CHOICE		EnumRange (OFF ON)
out.14.in.26	CHOICE		EnumRange (OFF ON)
out.14.in.27	CHOICE		EnumRange (OFF ON)
out.14.in.28	CHOICE		EnumRange (OFF ON)
out.14.in.29	CHOICE		EnumRange (OFF ON)
out.14.in.30	CHOICE		EnumRange (OFF ON)
out.14.in.31	CHOICE		EnumRange (OFF ON)
out.14.in.32	CHOICE		EnumRange (OFF ON)
out.14.in.33	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.14.in.34	CHOICE		EnumRange (OFF ON)
out.14.in.35	CHOICE		EnumRange (OFF ON)
out.14.in.36	CHOICE		EnumRange (OFF ON)
out.14.in.37	CHOICE		EnumRange (OFF ON)
out.14.in.38	CHOICE		EnumRange (OFF ON)
out.14.in.39	CHOICE		EnumRange (OFF ON)
out.14.in.40	CHOICE		EnumRange (OFF ON)
out.14.in.41	CHOICE		EnumRange (OFF ON)
out.14.in.42	CHOICE		EnumRange (OFF ON)
out.14.in.43	CHOICE		EnumRange (OFF ON)
out.14.in.44	CHOICE		EnumRange (OFF ON)
out.14.in.45	CHOICE		EnumRange (OFF ON)
out.14.in.46	CHOICE		EnumRange (OFF ON)
out.14.in.47	CHOICE		EnumRange (OFF ON)
out.14.in.48	CHOICE		EnumRange (OFF ON)
out.14.in.49	CHOICE		EnumRange (OFF ON)
out.14.in.50	CHOICE		EnumRange (OFF ON)
out.14.in.51	CHOICE		EnumRange (OFF ON)
out.14.in.52	CHOICE		EnumRange (OFF ON)
out.14.in.53	CHOICE		EnumRange (OFF ON)
out.14.in.54	CHOICE		EnumRange (OFF ON)
out.14.in.55	CHOICE		EnumRange (OFF ON)
out.14.in.56	CHOICE		EnumRange (OFF ON)
out.14.in.57	CHOICE		EnumRange (OFF ON)
out.14.in.58	CHOICE		EnumRange (OFF ON)
out.14.in.59	CHOICE		EnumRange (OFF ON)
out.14.in.60	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.14.in.61	CHOICE		EnumRange (OFF ON)
out.14.in.62	CHOICE		EnumRange (OFF ON)
out.14.in.63	CHOICE		EnumRange (OFF ON)
out.14.in.64	CHOICE		EnumRange (OFF ON)
out.15.in.01	CHOICE		EnumRange (OFF ON)
out.15.in.02	CHOICE		EnumRange (OFF ON)
out.15.in.03	CHOICE		EnumRange (OFF ON)
out.15.in.04	CHOICE		EnumRange (OFF ON)
out.15.in.05	CHOICE		EnumRange (OFF ON)
out.15.in.06	CHOICE		EnumRange (OFF ON)
out.15.in.07	CHOICE		EnumRange (OFF ON)
out.15.in.08	CHOICE		EnumRange (OFF ON)
out.15.in.09	CHOICE		EnumRange (OFF ON)
out.15.in.10	CHOICE		EnumRange (OFF ON)
out.15.in.11	CHOICE		EnumRange (OFF ON)
out.15.in.12	CHOICE		EnumRange (OFF ON)
out.15.in.13	CHOICE		EnumRange (OFF ON)
out.15.in.14	CHOICE		EnumRange (OFF ON)
out.15.in.15	CHOICE		EnumRange (OFF ON)
out.15.in.16	CHOICE		EnumRange (OFF ON)
out.15.in.17	CHOICE		EnumRange (OFF ON)
out.15.in.18	CHOICE		EnumRange (OFF ON)
out.15.in.19	CHOICE		EnumRange (OFF ON)
out.15.in.20	CHOICE		EnumRange (OFF ON)
out.15.in.21	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.15.in.22	CHOICE		EnumRange (OFF ON)
out.15.in.23	CHOICE		EnumRange (OFF ON)
out.15.in.24	CHOICE		EnumRange (OFF ON)
out.15.in.25	CHOICE		EnumRange (OFF ON)
out.15.in.26	CHOICE		EnumRange (OFF ON)
out.15.in.27	CHOICE		EnumRange (OFF ON)
out.15.in.28	CHOICE		EnumRange (OFF ON)
out.15.in.29	CHOICE		EnumRange (OFF ON)
out.15.in.30	CHOICE		EnumRange (OFF ON)
out.15.in.31	CHOICE		EnumRange (OFF ON)
out.15.in.32	CHOICE		EnumRange (OFF ON)
out.15.in.33	CHOICE		EnumRange (OFF ON)
out.15.in.34	CHOICE		EnumRange (OFF ON)
out.15.in.35	CHOICE		EnumRange (OFF ON)
out.15.in.36	CHOICE		EnumRange (OFF ON)
out.15.in.37	CHOICE		EnumRange (OFF ON)
out.15.in.38	CHOICE		EnumRange (OFF ON)
out.15.in.39	CHOICE		EnumRange (OFF ON)
out.15.in.40	CHOICE		EnumRange (OFF ON)
out.15.in.41	CHOICE		EnumRange (OFF ON)
out.15.in.42	CHOICE		EnumRange (OFF ON)
out.15.in.43	CHOICE		EnumRange (OFF ON)
out.15.in.44	CHOICE		EnumRange (OFF ON)
out.15.in.45	CHOICE		EnumRange (OFF ON)
out.15.in.46	CHOICE		EnumRange (OFF ON)
out.15.in.47	CHOICE		EnumRange (OFF ON)
out.15.in.48	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.15.in.49	CHOICE		EnumRange (OFF ON)
out.15.in.50	CHOICE		EnumRange (OFF ON)
out.15.in.51	CHOICE		EnumRange (OFF ON)
out.15.in.52	CHOICE		EnumRange (OFF ON)
out.15.in.53	CHOICE		EnumRange (OFF ON)
out.15.in.54	CHOICE		EnumRange (OFF ON)
out.15.in.55	CHOICE		EnumRange (OFF ON)
out.15.in.56	CHOICE		EnumRange (OFF ON)
out.15.in.57	CHOICE		EnumRange (OFF ON)
out.15.in.58	CHOICE		EnumRange (OFF ON)
out.15.in.59	CHOICE		EnumRange (OFF ON)
out.15.in.60	CHOICE		EnumRange (OFF ON)
out.15.in.61	CHOICE		EnumRange (OFF ON)
out.15.in.62	CHOICE		EnumRange (OFF ON)
out.15.in.63	CHOICE		EnumRange (OFF ON)
out.15.in.64	CHOICE		EnumRange (OFF ON)
out.16.in.01	CHOICE		EnumRange (OFF ON)
out.16.in.02	CHOICE		EnumRange (OFF ON)
out.16.in.03	CHOICE		EnumRange (OFF ON)
out.16.in.04	CHOICE		EnumRange (OFF ON)
out.16.in.05	CHOICE		EnumRange (OFF ON)
out.16.in.06	CHOICE		EnumRange (OFF ON)
out.16.in.07	CHOICE		EnumRange (OFF ON)
out.16.in.08	CHOICE		EnumRange (OFF ON)
out.16.in.09	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.16.in.10	CHOICE		EnumRange (OFF ON)
out.16.in.11	CHOICE		EnumRange (OFF ON)
out.16.in.12	CHOICE		EnumRange (OFF ON)
out.16.in.13	CHOICE		EnumRange (OFF ON)
out.16.in.14	CHOICE		EnumRange (OFF ON)
out.16.in.15	CHOICE		EnumRange (OFF ON)
out.16.in.16	CHOICE		EnumRange (OFF ON)
out.16.in.17	CHOICE		EnumRange (OFF ON)
out.16.in.18	CHOICE		EnumRange (OFF ON)
out.16.in.19	CHOICE		EnumRange (OFF ON)
out.16.in.20	CHOICE		EnumRange (OFF ON)
out.16.in.21	CHOICE		EnumRange (OFF ON)
out.16.in.22	CHOICE		EnumRange (OFF ON)
out.16.in.23	CHOICE		EnumRange (OFF ON)
out.16.in.24	CHOICE		EnumRange (OFF ON)
out.16.in.25	CHOICE		EnumRange (OFF ON)
out.16.in.26	CHOICE		EnumRange (OFF ON)
out.16.in.27	CHOICE		EnumRange (OFF ON)
out.16.in.28	CHOICE		EnumRange (OFF ON)
out.16.in.29	CHOICE		EnumRange (OFF ON)
out.16.in.30	CHOICE		EnumRange (OFF ON)
out.16.in.31	CHOICE		EnumRange (OFF ON)
out.16.in.32	CHOICE		EnumRange (OFF ON)
out.16.in.33	CHOICE		EnumRange (OFF ON)
out.16.in.34	CHOICE		EnumRange (OFF ON)
out.16.in.35	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.16.in.36	CHOICE		EnumRange (OFF ON)
out.16.in.37	CHOICE		EnumRange (OFF ON)
out.16.in.38	CHOICE		EnumRange (OFF ON)
out.16.in.39	CHOICE		EnumRange (OFF ON)
out.16.in.40	CHOICE		EnumRange (OFF ON)
out.16.in.41	CHOICE		EnumRange (OFF ON)
out.16.in.42	CHOICE		EnumRange (OFF ON)
out.16.in.43	CHOICE		EnumRange (OFF ON)
out.16.in.44	CHOICE		EnumRange (OFF ON)
out.16.in.45	CHOICE		EnumRange (OFF ON)
out.16.in.46	CHOICE		EnumRange (OFF ON)
out.16.in.47	CHOICE		EnumRange (OFF ON)
out.16.in.48	CHOICE		EnumRange (OFF ON)
out.16.in.49	CHOICE		EnumRange (OFF ON)
out.16.in.50	CHOICE		EnumRange (OFF ON)
out.16.in.51	CHOICE		EnumRange (OFF ON)
out.16.in.52	CHOICE		EnumRange (OFF ON)
out.16.in.53	CHOICE		EnumRange (OFF ON)
out.16.in.54	CHOICE		EnumRange (OFF ON)
out.16.in.55	CHOICE		EnumRange (OFF ON)
out.16.in.56	CHOICE		EnumRange (OFF ON)
out.16.in.57	CHOICE		EnumRange (OFF ON)
out.16.in.58	CHOICE		EnumRange (OFF ON)
out.16.in.59	CHOICE		EnumRange (OFF ON)
out.16.in.60	CHOICE		EnumRange (OFF ON)
out.16.in.61	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.16.in.62	CHOICE		EnumRange (OFF ON)
out.16.in.63	CHOICE		EnumRange (OFF ON)
out.16.in.64	CHOICE		EnumRange (OFF ON)
out.17.in.01	CHOICE		EnumRange (OFF ON)
out.17.in.02	CHOICE		EnumRange (OFF ON)
out.17.in.03	CHOICE		EnumRange (OFF ON)
out.17.in.04	CHOICE		EnumRange (OFF ON)
out.17.in.05	CHOICE		EnumRange (OFF ON)
out.17.in.06	CHOICE		EnumRange (OFF ON)
out.17.in.07	CHOICE		EnumRange (OFF ON)
out.17.in.08	CHOICE		EnumRange (OFF ON)
out.17.in.09	CHOICE		EnumRange (OFF ON)
out.17.in.10	CHOICE		EnumRange (OFF ON)
out.17.in.11	CHOICE		EnumRange (OFF ON)
out.17.in.12	CHOICE		EnumRange (OFF ON)
out.17.in.13	CHOICE		EnumRange (OFF ON)
out.17.in.14	CHOICE		EnumRange (OFF ON)
out.17.in.15	CHOICE		EnumRange (OFF ON)
out.17.in.16	CHOICE		EnumRange (OFF ON)
out.17.in.17	CHOICE		EnumRange (OFF ON)
out.17.in.18	CHOICE		EnumRange (OFF ON)
out.17.in.19	CHOICE		EnumRange (OFF ON)
out.17.in.20	CHOICE		EnumRange (OFF ON)
out.17.in.21	CHOICE		EnumRange (OFF ON)
out.17.in.22	CHOICE		EnumRange (OFF ON)
out.17.in.23	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.17.in.24	CHOICE		EnumRange (OFF ON)
out.17.in.25	CHOICE		EnumRange (OFF ON)
out.17.in.26	CHOICE		EnumRange (OFF ON)
out.17.in.27	CHOICE		EnumRange (OFF ON)
out.17.in.28	CHOICE		EnumRange (OFF ON)
out.17.in.29	CHOICE		EnumRange (OFF ON)
out.17.in.30	CHOICE		EnumRange (OFF ON)
out.17.in.31	CHOICE		EnumRange (OFF ON)
out.17.in.32	CHOICE		EnumRange (OFF ON)
out.17.in.33	CHOICE		EnumRange (OFF ON)
out.17.in.34	CHOICE		EnumRange (OFF ON)
out.17.in.35	CHOICE		EnumRange (OFF ON)
out.17.in.36	CHOICE		EnumRange (OFF ON)
out.17.in.37	CHOICE		EnumRange (OFF ON)
out.17.in.38	CHOICE		EnumRange (OFF ON)
out.17.in.39	CHOICE		EnumRange (OFF ON)
out.17.in.40	CHOICE		EnumRange (OFF ON)
out.17.in.41	CHOICE		EnumRange (OFF ON)
out.17.in.42	CHOICE		EnumRange (OFF ON)
out.17.in.43	CHOICE		EnumRange (OFF ON)
out.17.in.44	CHOICE		EnumRange (OFF ON)
out.17.in.45	CHOICE		EnumRange (OFF ON)
out.17.in.46	CHOICE		EnumRange (OFF ON)
out.17.in.47	CHOICE		EnumRange (OFF ON)
out.17.in.48	CHOICE		EnumRange (OFF ON)
out.17.in.49	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.17.in.50	CHOICE		EnumRange (OFF ON)
out.17.in.51	CHOICE		EnumRange (OFF ON)
out.17.in.52	CHOICE		EnumRange (OFF ON)
out.17.in.53	CHOICE		EnumRange (OFF ON)
out.17.in.54	CHOICE		EnumRange (OFF ON)
out.17.in.55	CHOICE		EnumRange (OFF ON)
out.17.in.56	CHOICE		EnumRange (OFF ON)
out.17.in.57	CHOICE		EnumRange (OFF ON)
out.17.in.58	CHOICE		EnumRange (OFF ON)
out.17.in.59	CHOICE		EnumRange (OFF ON)
out.17.in.60	CHOICE		EnumRange (OFF ON)
out.17.in.61	CHOICE		EnumRange (OFF ON)
out.17.in.62	CHOICE		EnumRange (OFF ON)
out.17.in.63	CHOICE		EnumRange (OFF ON)
out.17.in.64	CHOICE		EnumRange (OFF ON)
out.18.in.01	CHOICE		EnumRange (OFF ON)
out.18.in.02	CHOICE		EnumRange (OFF ON)
out.18.in.03	CHOICE		EnumRange (OFF ON)
out.18.in.04	CHOICE		EnumRange (OFF ON)
out.18.in.05	CHOICE		EnumRange (OFF ON)
out.18.in.06	CHOICE		EnumRange (OFF ON)
out.18.in.07	CHOICE		EnumRange (OFF ON)
out.18.in.08	CHOICE		EnumRange (OFF ON)
out.18.in.09	CHOICE		EnumRange (OFF ON)
out.18.in.10	CHOICE		EnumRange (OFF ON)
out.18.in.11	CHOICE		EnumRange (OFF ON)
out.18.in.12	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.18.in.13	CHOICE		EnumRange (OFF ON)
out.18.in.14	CHOICE		EnumRange (OFF ON)
out.18.in.15	CHOICE		EnumRange (OFF ON)
out.18.in.16	CHOICE		EnumRange (OFF ON)
out.18.in.17	CHOICE		EnumRange (OFF ON)
out.18.in.18	CHOICE		EnumRange (OFF ON)
out.18.in.19	CHOICE		EnumRange (OFF ON)
out.18.in.20	CHOICE		EnumRange (OFF ON)
out.18.in.21	CHOICE		EnumRange (OFF ON)
out.18.in.22	CHOICE		EnumRange (OFF ON)
out.18.in.23	CHOICE		EnumRange (OFF ON)
out.18.in.24	CHOICE		EnumRange (OFF ON)
out.18.in.25	CHOICE		EnumRange (OFF ON)
out.18.in.26	CHOICE		EnumRange (OFF ON)
out.18.in.27	CHOICE		EnumRange (OFF ON)
out.18.in.28	CHOICE		EnumRange (OFF ON)
out.18.in.29	CHOICE		EnumRange (OFF ON)
out.18.in.30	CHOICE		EnumRange (OFF ON)
out.18.in.31	CHOICE		EnumRange (OFF ON)
out.18.in.32	CHOICE		EnumRange (OFF ON)
out.18.in.33	CHOICE		EnumRange (OFF ON)
out.18.in.34	CHOICE		EnumRange (OFF ON)
out.18.in.35	CHOICE		EnumRange (OFF ON)
out.18.in.36	CHOICE		EnumRange (OFF ON)
out.18.in.37	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.18.in.38	CHOICE		EnumRange (OFF ON)
out.18.in.39	CHOICE		EnumRange (OFF ON)
out.18.in.40	CHOICE		EnumRange (OFF ON)
out.18.in.41	CHOICE		EnumRange (OFF ON)
out.18.in.42	CHOICE		EnumRange (OFF ON)
out.18.in.43	CHOICE		EnumRange (OFF ON)
out.18.in.44	CHOICE		EnumRange (OFF ON)
out.18.in.45	CHOICE		EnumRange (OFF ON)
out.18.in.46	CHOICE		EnumRange (OFF ON)
out.18.in.47	CHOICE		EnumRange (OFF ON)
out.18.in.48	CHOICE		EnumRange (OFF ON)
out.18.in.49	CHOICE		EnumRange (OFF ON)
out.18.in.50	CHOICE		EnumRange (OFF ON)
out.18.in.51	CHOICE		EnumRange (OFF ON)
out.18.in.52	CHOICE		EnumRange (OFF ON)
out.18.in.53	CHOICE		EnumRange (OFF ON)
out.18.in.54	CHOICE		EnumRange (OFF ON)
out.18.in.55	CHOICE		EnumRange (OFF ON)
out.18.in.56	CHOICE		EnumRange (OFF ON)
out.18.in.57	CHOICE		EnumRange (OFF ON)
out.18.in.58	CHOICE		EnumRange (OFF ON)
out.18.in.59	CHOICE		EnumRange (OFF ON)
out.18.in.60	CHOICE		EnumRange (OFF ON)
out.18.in.61	CHOICE		EnumRange (OFF ON)
out.18.in.62	CHOICE		EnumRange (OFF ON)
out.18.in.63	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.18.in.64	CHOICE		EnumRange (OFF ON)
out.19.in.01	CHOICE		EnumRange (OFF ON)
out.19.in.02	CHOICE		EnumRange (OFF ON)
out.19.in.03	CHOICE		EnumRange (OFF ON)
out.19.in.04	CHOICE		EnumRange (OFF ON)
out.19.in.05	CHOICE		EnumRange (OFF ON)
out.19.in.06	CHOICE		EnumRange (OFF ON)
out.19.in.07	CHOICE		EnumRange (OFF ON)
out.19.in.08	CHOICE		EnumRange (OFF ON)
out.19.in.09	CHOICE		EnumRange (OFF ON)
out.19.in.10	CHOICE		EnumRange (OFF ON)
out.19.in.11	CHOICE		EnumRange (OFF ON)
out.19.in.12	CHOICE		EnumRange (OFF ON)
out.19.in.13	CHOICE		EnumRange (OFF ON)
out.19.in.14	CHOICE		EnumRange (OFF ON)
out.19.in.15	CHOICE		EnumRange (OFF ON)
out.19.in.16	CHOICE		EnumRange (OFF ON)
out.19.in.17	CHOICE		EnumRange (OFF ON)
out.19.in.18	CHOICE		EnumRange (OFF ON)
out.19.in.19	CHOICE		EnumRange (OFF ON)
out.19.in.20	CHOICE		EnumRange (OFF ON)
out.19.in.21	CHOICE		EnumRange (OFF ON)
out.19.in.22	CHOICE		EnumRange (OFF ON)
out.19.in.23	CHOICE		EnumRange (OFF ON)
out.19.in.24	CHOICE		EnumRange (OFF ON)
out.19.in.25	CHOICE		EnumRange (OFF ON)
out.19.in.26	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.19.in.27	CHOICE		EnumRange (OFF ON)
out.19.in.28	CHOICE		EnumRange (OFF ON)
out.19.in.29	CHOICE		EnumRange (OFF ON)
out.19.in.30	CHOICE		EnumRange (OFF ON)
out.19.in.31	CHOICE		EnumRange (OFF ON)
out.19.in.32	CHOICE		EnumRange (OFF ON)
out.19.in.33	CHOICE		EnumRange (OFF ON)
out.19.in.34	CHOICE		EnumRange (OFF ON)
out.19.in.35	CHOICE		EnumRange (OFF ON)
out.19.in.36	CHOICE		EnumRange (OFF ON)
out.19.in.37	CHOICE		EnumRange (OFF ON)
out.19.in.38	CHOICE		EnumRange (OFF ON)
out.19.in.39	CHOICE		EnumRange (OFF ON)
out.19.in.40	CHOICE		EnumRange (OFF ON)
out.19.in.41	CHOICE		EnumRange (OFF ON)
out.19.in.42	CHOICE		EnumRange (OFF ON)
out.19.in.43	CHOICE		EnumRange (OFF ON)
out.19.in.44	CHOICE		EnumRange (OFF ON)
out.19.in.45	CHOICE		EnumRange (OFF ON)
out.19.in.46	CHOICE		EnumRange (OFF ON)
out.19.in.47	CHOICE		EnumRange (OFF ON)
out.19.in.48	CHOICE		EnumRange (OFF ON)
out.19.in.49	CHOICE		EnumRange (OFF ON)
out.19.in.50	CHOICE		EnumRange (OFF ON)
out.19.in.51	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.19.in.52	CHOICE		EnumRange (OFF ON)
out.19.in.53	CHOICE		EnumRange (OFF ON)
out.19.in.54	CHOICE		EnumRange (OFF ON)
out.19.in.55	CHOICE		EnumRange (OFF ON)
out.19.in.56	CHOICE		EnumRange (OFF ON)
out.19.in.57	CHOICE		EnumRange (OFF ON)
out.19.in.58	CHOICE		EnumRange (OFF ON)
out.19.in.59	CHOICE		EnumRange (OFF ON)
out.19.in.60	CHOICE		EnumRange (OFF ON)
out.19.in.61	CHOICE		EnumRange (OFF ON)
out.19.in.62	CHOICE		EnumRange (OFF ON)
out.19.in.63	CHOICE		EnumRange (OFF ON)
out.19.in.64	CHOICE		EnumRange (OFF ON)
out.20.in.01	CHOICE		EnumRange (OFF ON)
out.20.in.02	CHOICE		EnumRange (OFF ON)
out.20.in.03	CHOICE		EnumRange (OFF ON)
out.20.in.04	CHOICE		EnumRange (OFF ON)
out.20.in.05	CHOICE		EnumRange (OFF ON)
out.20.in.06	CHOICE		EnumRange (OFF ON)
out.20.in.07	CHOICE		EnumRange (OFF ON)
out.20.in.08	CHOICE		EnumRange (OFF ON)
out.20.in.09	CHOICE		EnumRange (OFF ON)
out.20.in.10	CHOICE		EnumRange (OFF ON)
out.20.in.11	CHOICE		EnumRange (OFF ON)
out.20.in.12	CHOICE		EnumRange (OFF ON)
out.20.in.13	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.20.in.14	CHOICE		EnumRange (OFF ON)
out.20.in.15	CHOICE		EnumRange (OFF ON)
out.20.in.16	CHOICE		EnumRange (OFF ON)
out.20.in.17	CHOICE		EnumRange (OFF ON)
out.20.in.18	CHOICE		EnumRange (OFF ON)
out.20.in.19	CHOICE		EnumRange (OFF ON)
out.20.in.20	CHOICE		EnumRange (OFF ON)
out.20.in.21	CHOICE		EnumRange (OFF ON)
out.20.in.22	CHOICE		EnumRange (OFF ON)
out.20.in.23	CHOICE		EnumRange (OFF ON)
out.20.in.24	CHOICE		EnumRange (OFF ON)
out.20.in.25	CHOICE		EnumRange (OFF ON)
out.20.in.26	CHOICE		EnumRange (OFF ON)
out.20.in.27	CHOICE		EnumRange (OFF ON)
out.20.in.28	CHOICE		EnumRange (OFF ON)
out.20.in.29	CHOICE		EnumRange (OFF ON)
out.20.in.30	CHOICE		EnumRange (OFF ON)
out.20.in.31	CHOICE		EnumRange (OFF ON)
out.20.in.32	CHOICE		EnumRange (OFF ON)
out.20.in.33	CHOICE		EnumRange (OFF ON)
out.20.in.34	CHOICE		EnumRange (OFF ON)
out.20.in.35	CHOICE		EnumRange (OFF ON)
out.20.in.36	CHOICE		EnumRange (OFF ON)
out.20.in.37	CHOICE		EnumRange (OFF ON)
out.20.in.38	CHOICE		EnumRange (OFF ON)
out.20.in.39	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.20.in.40	CHOICE		EnumRange (OFF ON)
out.20.in.41	CHOICE		EnumRange (OFF ON)
out.20.in.42	CHOICE		EnumRange (OFF ON)
out.20.in.43	CHOICE		EnumRange (OFF ON)
out.20.in.44	CHOICE		EnumRange (OFF ON)
out.20.in.45	CHOICE		EnumRange (OFF ON)
out.20.in.46	CHOICE		EnumRange (OFF ON)
out.20.in.47	CHOICE		EnumRange (OFF ON)
out.20.in.48	CHOICE		EnumRange (OFF ON)
out.20.in.49	CHOICE		EnumRange (OFF ON)
out.20.in.50	CHOICE		EnumRange (OFF ON)
out.20.in.51	CHOICE		EnumRange (OFF ON)
out.20.in.52	CHOICE		EnumRange (OFF ON)
out.20.in.53	CHOICE		EnumRange (OFF ON)
out.20.in.54	CHOICE		EnumRange (OFF ON)
out.20.in.55	CHOICE		EnumRange (OFF ON)
out.20.in.56	CHOICE		EnumRange (OFF ON)
out.20.in.57	CHOICE		EnumRange (OFF ON)
out.20.in.58	CHOICE		EnumRange (OFF ON)
out.20.in.59	CHOICE		EnumRange (OFF ON)
out.20.in.60	CHOICE		EnumRange (OFF ON)
out.20.in.61	CHOICE		EnumRange (OFF ON)
out.20.in.62	CHOICE		EnumRange (OFF ON)
out.20.in.63	CHOICE		EnumRange (OFF ON)
out.20.in.64	CHOICE		EnumRange (OFF ON)
out.21.in.01	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.21.in.02	CHOICE		EnumRange (OFF ON)
out.21.in.03	CHOICE		EnumRange (OFF ON)
out.21.in.04	CHOICE		EnumRange (OFF ON)
out.21.in.05	CHOICE		EnumRange (OFF ON)
out.21.in.06	CHOICE		EnumRange (OFF ON)
out.21.in.07	CHOICE		EnumRange (OFF ON)
out.21.in.08	CHOICE		EnumRange (OFF ON)
out.21.in.09	CHOICE		EnumRange (OFF ON)
out.21.in.10	CHOICE		EnumRange (OFF ON)
out.21.in.11	CHOICE		EnumRange (OFF ON)
out.21.in.12	CHOICE		EnumRange (OFF ON)
out.21.in.13	CHOICE		EnumRange (OFF ON)
out.21.in.14	CHOICE		EnumRange (OFF ON)
out.21.in.15	CHOICE		EnumRange (OFF ON)
out.21.in.16	CHOICE		EnumRange (OFF ON)
out.21.in.17	CHOICE		EnumRange (OFF ON)
out.21.in.18	CHOICE		EnumRange (OFF ON)
out.21.in.19	CHOICE		EnumRange (OFF ON)
out.21.in.20	CHOICE		EnumRange (OFF ON)
out.21.in.21	CHOICE		EnumRange (OFF ON)
out.21.in.22	CHOICE		EnumRange (OFF ON)
out.21.in.23	CHOICE		EnumRange (OFF ON)
out.21.in.24	CHOICE		EnumRange (OFF ON)
out.21.in.25	CHOICE		EnumRange (OFF ON)
out.21.in.26	CHOICE		EnumRange (OFF ON)
out.21.in.27	CHOICE		EnumRange (OFF ON)
out.21.in.28	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.21.in.29	CHOICE		EnumRange (OFF ON)
out.21.in.30	CHOICE		EnumRange (OFF ON)
out.21.in.31	CHOICE		EnumRange (OFF ON)
out.21.in.32	CHOICE		EnumRange (OFF ON)
out.21.in.33	CHOICE		EnumRange (OFF ON)
out.21.in.34	CHOICE		EnumRange (OFF ON)
out.21.in.35	CHOICE		EnumRange (OFF ON)
out.21.in.36	CHOICE		EnumRange (OFF ON)
out.21.in.37	CHOICE		EnumRange (OFF ON)
out.21.in.38	CHOICE		EnumRange (OFF ON)
out.21.in.39	CHOICE		EnumRange (OFF ON)
out.21.in.40	CHOICE		EnumRange (OFF ON)
out.21.in.41	CHOICE		EnumRange (OFF ON)
out.21.in.42	CHOICE		EnumRange (OFF ON)
out.21.in.43	CHOICE		EnumRange (OFF ON)
out.21.in.44	CHOICE		EnumRange (OFF ON)
out.21.in.45	CHOICE		EnumRange (OFF ON)
out.21.in.46	CHOICE		EnumRange (OFF ON)
out.21.in.47	CHOICE		EnumRange (OFF ON)
out.21.in.48	CHOICE		EnumRange (OFF ON)
out.21.in.49	CHOICE		EnumRange (OFF ON)
out.21.in.50	CHOICE		EnumRange (OFF ON)
out.21.in.51	CHOICE		EnumRange (OFF ON)
out.21.in.52	CHOICE		EnumRange (OFF ON)
out.21.in.53	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.21.in.54	CHOICE		EnumRange (OFF ON)
out.21.in.55	CHOICE		EnumRange (OFF ON)
out.21.in.56	CHOICE		EnumRange (OFF ON)
out.21.in.57	CHOICE		EnumRange (OFF ON)
out.21.in.58	CHOICE		EnumRange (OFF ON)
out.21.in.59	CHOICE		EnumRange (OFF ON)
out.21.in.60	CHOICE		EnumRange (OFF ON)
out.21.in.61	CHOICE		EnumRange (OFF ON)
out.21.in.62	CHOICE		EnumRange (OFF ON)
out.21.in.63	CHOICE		EnumRange (OFF ON)
out.21.in.64	CHOICE		EnumRange (OFF ON)
out.22.in.01	CHOICE		EnumRange (OFF ON)
out.22.in.02	CHOICE		EnumRange (OFF ON)
out.22.in.03	CHOICE		EnumRange (OFF ON)
out.22.in.04	CHOICE		EnumRange (OFF ON)
out.22.in.05	CHOICE		EnumRange (OFF ON)
out.22.in.06	CHOICE		EnumRange (OFF ON)
out.22.in.07	CHOICE		EnumRange (OFF ON)
out.22.in.08	CHOICE		EnumRange (OFF ON)
out.22.in.09	CHOICE		EnumRange (OFF ON)
out.22.in.10	CHOICE		EnumRange (OFF ON)
out.22.in.11	CHOICE		EnumRange (OFF ON)
out.22.in.12	CHOICE		EnumRange (OFF ON)
out.22.in.13	CHOICE		EnumRange (OFF ON)
out.22.in.14	CHOICE		EnumRange (OFF ON)
out.22.in.15	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.22.in.16	CHOICE		EnumRange (OFF ON)
out.22.in.17	CHOICE		EnumRange (OFF ON)
out.22.in.18	CHOICE		EnumRange (OFF ON)
out.22.in.19	CHOICE		EnumRange (OFF ON)
out.22.in.20	CHOICE		EnumRange (OFF ON)
out.22.in.21	CHOICE		EnumRange (OFF ON)
out.22.in.22	CHOICE		EnumRange (OFF ON)
out.22.in.23	CHOICE		EnumRange (OFF ON)
out.22.in.24	CHOICE		EnumRange (OFF ON)
out.22.in.25	CHOICE		EnumRange (OFF ON)
out.22.in.26	CHOICE		EnumRange (OFF ON)
out.22.in.27	CHOICE		EnumRange (OFF ON)
out.22.in.28	CHOICE		EnumRange (OFF ON)
out.22.in.29	CHOICE		EnumRange (OFF ON)
out.22.in.30	CHOICE		EnumRange (OFF ON)
out.22.in.31	CHOICE		EnumRange (OFF ON)
out.22.in.32	CHOICE		EnumRange (OFF ON)
out.22.in.33	CHOICE		EnumRange (OFF ON)
out.22.in.34	CHOICE		EnumRange (OFF ON)
out.22.in.35	CHOICE		EnumRange (OFF ON)
out.22.in.36	CHOICE		EnumRange (OFF ON)
out.22.in.37	CHOICE		EnumRange (OFF ON)
out.22.in.38	CHOICE		EnumRange (OFF ON)
out.22.in.39	CHOICE		EnumRange (OFF ON)
out.22.in.40	CHOICE		EnumRange (OFF ON)
out.22.in.41	CHOICE		EnumRange (OFF ON)
out.22.in.42	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.22.in.43	CHOICE		EnumRange (OFF ON)
out.22.in.44	CHOICE		EnumRange (OFF ON)
out.22.in.45	CHOICE		EnumRange (OFF ON)
out.22.in.46	CHOICE		EnumRange (OFF ON)
out.22.in.47	CHOICE		EnumRange (OFF ON)
out.22.in.48	CHOICE		EnumRange (OFF ON)
out.22.in.49	CHOICE		EnumRange (OFF ON)
out.22.in.50	CHOICE		EnumRange (OFF ON)
out.22.in.51	CHOICE		EnumRange (OFF ON)
out.22.in.52	CHOICE		EnumRange (OFF ON)
out.22.in.53	CHOICE		EnumRange (OFF ON)
out.22.in.54	CHOICE		EnumRange (OFF ON)
out.22.in.55	CHOICE		EnumRange (OFF ON)
out.22.in.56	CHOICE		EnumRange (OFF ON)
out.22.in.57	CHOICE		EnumRange (OFF ON)
out.22.in.58	CHOICE		EnumRange (OFF ON)
out.22.in.59	CHOICE		EnumRange (OFF ON)
out.22.in.60	CHOICE		EnumRange (OFF ON)
out.22.in.61	CHOICE		EnumRange (OFF ON)
out.22.in.62	CHOICE		EnumRange (OFF ON)
out.22.in.63	CHOICE		EnumRange (OFF ON)
out.22.in.64	CHOICE		EnumRange (OFF ON)
out.23.in.01	CHOICE		EnumRange (OFF ON)
out.23.in.02	CHOICE		EnumRange (OFF ON)
out.23.in.03	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.23.in.04	CHOICE		EnumRange (OFF ON)
out.23.in.05	CHOICE		EnumRange (OFF ON)
out.23.in.06	CHOICE		EnumRange (OFF ON)
out.23.in.07	CHOICE		EnumRange (OFF ON)
out.23.in.08	CHOICE		EnumRange (OFF ON)
out.23.in.09	CHOICE		EnumRange (OFF ON)
out.23.in.10	CHOICE		EnumRange (OFF ON)
out.23.in.11	CHOICE		EnumRange (OFF ON)
out.23.in.12	CHOICE		EnumRange (OFF ON)
out.23.in.13	CHOICE		EnumRange (OFF ON)
out.23.in.14	CHOICE		EnumRange (OFF ON)
out.23.in.15	CHOICE		EnumRange (OFF ON)
out.23.in.16	CHOICE		EnumRange (OFF ON)
out.23.in.17	CHOICE		EnumRange (OFF ON)
out.23.in.18	CHOICE		EnumRange (OFF ON)
out.23.in.19	CHOICE		EnumRange (OFF ON)
out.23.in.20	CHOICE		EnumRange (OFF ON)
out.23.in.21	CHOICE		EnumRange (OFF ON)
out.23.in.22	CHOICE		EnumRange (OFF ON)
out.23.in.23	CHOICE		EnumRange (OFF ON)
out.23.in.24	CHOICE		EnumRange (OFF ON)
out.23.in.25	CHOICE		EnumRange (OFF ON)
out.23.in.26	CHOICE		EnumRange (OFF ON)
out.23.in.27	CHOICE		EnumRange (OFF ON)
out.23.in.28	CHOICE		EnumRange (OFF ON)
out.23.in.29	CHOICE		EnumRange (OFF ON)
out.23.in.30	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.23.in.31	CHOICE		EnumRange (OFF ON)
out.23.in.32	CHOICE		EnumRange (OFF ON)
out.23.in.33	CHOICE		EnumRange (OFF ON)
out.23.in.34	CHOICE		EnumRange (OFF ON)
out.23.in.35	CHOICE		EnumRange (OFF ON)
out.23.in.36	CHOICE		EnumRange (OFF ON)
out.23.in.37	CHOICE		EnumRange (OFF ON)
out.23.in.38	CHOICE		EnumRange (OFF ON)
out.23.in.39	CHOICE		EnumRange (OFF ON)
out.23.in.40	CHOICE		EnumRange (OFF ON)
out.23.in.41	CHOICE		EnumRange (OFF ON)
out.23.in.42	CHOICE		EnumRange (OFF ON)
out.23.in.43	CHOICE		EnumRange (OFF ON)
out.23.in.44	CHOICE		EnumRange (OFF ON)
out.23.in.45	CHOICE		EnumRange (OFF ON)
out.23.in.46	CHOICE		EnumRange (OFF ON)
out.23.in.47	CHOICE		EnumRange (OFF ON)
out.23.in.48	CHOICE		EnumRange (OFF ON)
out.23.in.49	CHOICE		EnumRange (OFF ON)
out.23.in.50	CHOICE		EnumRange (OFF ON)
out.23.in.51	CHOICE		EnumRange (OFF ON)
out.23.in.52	CHOICE		EnumRange (OFF ON)
out.23.in.53	CHOICE		EnumRange (OFF ON)
out.23.in.54	CHOICE		EnumRange (OFF ON)
out.23.in.55	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.23.in.56	CHOICE		EnumRange (OFF ON)
out.23.in.57	CHOICE		EnumRange (OFF ON)
out.23.in.58	CHOICE		EnumRange (OFF ON)
out.23.in.59	CHOICE		EnumRange (OFF ON)
out.23.in.60	CHOICE		EnumRange (OFF ON)
out.23.in.61	CHOICE		EnumRange (OFF ON)
out.23.in.62	CHOICE		EnumRange (OFF ON)
out.23.in.63	CHOICE		EnumRange (OFF ON)
out.23.in.64	CHOICE		EnumRange (OFF ON)
out.24.in.01	CHOICE		EnumRange (OFF ON)
out.24.in.02	CHOICE		EnumRange (OFF ON)
out.24.in.03	CHOICE		EnumRange (OFF ON)
out.24.in.04	CHOICE		EnumRange (OFF ON)
out.24.in.05	CHOICE		EnumRange (OFF ON)
out.24.in.06	CHOICE		EnumRange (OFF ON)
out.24.in.07	CHOICE		EnumRange (OFF ON)
out.24.in.08	CHOICE		EnumRange (OFF ON)
out.24.in.09	CHOICE		EnumRange (OFF ON)
out.24.in.10	CHOICE		EnumRange (OFF ON)
out.24.in.11	CHOICE		EnumRange (OFF ON)
out.24.in.12	CHOICE		EnumRange (OFF ON)
out.24.in.13	CHOICE		EnumRange (OFF ON)
out.24.in.14	CHOICE		EnumRange (OFF ON)
out.24.in.15	CHOICE		EnumRange (OFF ON)
out.24.in.16	CHOICE		EnumRange (OFF ON)
out.24.in.17	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.24.in.18	CHOICE		EnumRange (OFF ON)
out.24.in.19	CHOICE		EnumRange (OFF ON)
out.24.in.20	CHOICE		EnumRange (OFF ON)
out.24.in.21	CHOICE		EnumRange (OFF ON)
out.24.in.22	CHOICE		EnumRange (OFF ON)
out.24.in.23	CHOICE		EnumRange (OFF ON)
out.24.in.24	CHOICE		EnumRange (OFF ON)
out.24.in.25	CHOICE		EnumRange (OFF ON)
out.24.in.26	CHOICE		EnumRange (OFF ON)
out.24.in.27	CHOICE		EnumRange (OFF ON)
out.24.in.28	CHOICE		EnumRange (OFF ON)
out.24.in.29	CHOICE		EnumRange (OFF ON)
out.24.in.30	CHOICE		EnumRange (OFF ON)
out.24.in.31	CHOICE		EnumRange (OFF ON)
out.24.in.32	CHOICE		EnumRange (OFF ON)
out.24.in.33	CHOICE		EnumRange (OFF ON)
out.24.in.34	CHOICE		EnumRange (OFF ON)
out.24.in.35	CHOICE		EnumRange (OFF ON)
out.24.in.36	CHOICE		EnumRange (OFF ON)
out.24.in.37	CHOICE		EnumRange (OFF ON)
out.24.in.38	CHOICE		EnumRange (OFF ON)
out.24.in.39	CHOICE		EnumRange (OFF ON)
out.24.in.40	CHOICE		EnumRange (OFF ON)
out.24.in.41	CHOICE		EnumRange (OFF ON)
out.24.in.42	CHOICE		EnumRange (OFF ON)
out.24.in.43	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.24.in.44	CHOICE		EnumRange (OFF ON)
out.24.in.45	CHOICE		EnumRange (OFF ON)
out.24.in.46	CHOICE		EnumRange (OFF ON)
out.24.in.47	CHOICE		EnumRange (OFF ON)
out.24.in.48	CHOICE		EnumRange (OFF ON)
out.24.in.49	CHOICE		EnumRange (OFF ON)
out.24.in.50	CHOICE		EnumRange (OFF ON)
out.24.in.51	CHOICE		EnumRange (OFF ON)
out.24.in.52	CHOICE		EnumRange (OFF ON)
out.24.in.53	CHOICE		EnumRange (OFF ON)
out.24.in.54	CHOICE		EnumRange (OFF ON)
out.24.in.55	CHOICE		EnumRange (OFF ON)
out.24.in.56	CHOICE		EnumRange (OFF ON)
out.24.in.57	CHOICE		EnumRange (OFF ON)
out.24.in.58	CHOICE		EnumRange (OFF ON)
out.24.in.59	CHOICE		EnumRange (OFF ON)
out.24.in.60	CHOICE		EnumRange (OFF ON)
out.24.in.61	CHOICE		EnumRange (OFF ON)
out.24.in.62	CHOICE		EnumRange (OFF ON)
out.24.in.63	CHOICE		EnumRange (OFF ON)
out.24.in.64	CHOICE		EnumRange (OFF ON)
out.25.in.01	CHOICE		EnumRange (OFF ON)
out.25.in.02	CHOICE		EnumRange (OFF ON)
out.25.in.03	CHOICE		EnumRange (OFF ON)
out.25.in.04	CHOICE		EnumRange (OFF ON)
out.25.in.05	CHOICE		EnumRange (OFF ON)
out.25.in.06	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.25.in.07	CHOICE		EnumRange (OFF ON)
out.25.in.08	CHOICE		EnumRange (OFF ON)
out.25.in.09	CHOICE		EnumRange (OFF ON)
out.25.in.10	CHOICE		EnumRange (OFF ON)
out.25.in.11	CHOICE		EnumRange (OFF ON)
out.25.in.12	CHOICE		EnumRange (OFF ON)
out.25.in.13	CHOICE		EnumRange (OFF ON)
out.25.in.14	CHOICE		EnumRange (OFF ON)
out.25.in.15	CHOICE		EnumRange (OFF ON)
out.25.in.16	CHOICE		EnumRange (OFF ON)
out.25.in.17	CHOICE		EnumRange (OFF ON)
out.25.in.18	CHOICE		EnumRange (OFF ON)
out.25.in.19	CHOICE		EnumRange (OFF ON)
out.25.in.20	CHOICE		EnumRange (OFF ON)
out.25.in.21	CHOICE		EnumRange (OFF ON)
out.25.in.22	CHOICE		EnumRange (OFF ON)
out.25.in.23	CHOICE		EnumRange (OFF ON)
out.25.in.24	CHOICE		EnumRange (OFF ON)
out.25.in.25	CHOICE		EnumRange (OFF ON)
out.25.in.26	CHOICE		EnumRange (OFF ON)
out.25.in.27	CHOICE		EnumRange (OFF ON)
out.25.in.28	CHOICE		EnumRange (OFF ON)
out.25.in.29	CHOICE		EnumRange (OFF ON)
out.25.in.30	CHOICE		EnumRange (OFF ON)
out.25.in.31	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.25.in.32	CHOICE		EnumRange (OFF ON)
out.25.in.33	CHOICE		EnumRange (OFF ON)
out.25.in.34	CHOICE		EnumRange (OFF ON)
out.25.in.35	CHOICE		EnumRange (OFF ON)
out.25.in.36	CHOICE		EnumRange (OFF ON)
out.25.in.37	CHOICE		EnumRange (OFF ON)
out.25.in.38	CHOICE		EnumRange (OFF ON)
out.25.in.39	CHOICE		EnumRange (OFF ON)
out.25.in.40	CHOICE		EnumRange (OFF ON)
out.25.in.41	CHOICE		EnumRange (OFF ON)
out.25.in.42	CHOICE		EnumRange (OFF ON)
out.25.in.43	CHOICE		EnumRange (OFF ON)
out.25.in.44	CHOICE		EnumRange (OFF ON)
out.25.in.45	CHOICE		EnumRange (OFF ON)
out.25.in.46	CHOICE		EnumRange (OFF ON)
out.25.in.47	CHOICE		EnumRange (OFF ON)
out.25.in.48	CHOICE		EnumRange (OFF ON)
out.25.in.49	CHOICE		EnumRange (OFF ON)
out.25.in.50	CHOICE		EnumRange (OFF ON)
out.25.in.51	CHOICE		EnumRange (OFF ON)
out.25.in.52	CHOICE		EnumRange (OFF ON)
out.25.in.53	CHOICE		EnumRange (OFF ON)
out.25.in.54	CHOICE		EnumRange (OFF ON)
out.25.in.55	CHOICE		EnumRange (OFF ON)
out.25.in.56	CHOICE		EnumRange (OFF ON)
out.25.in.57	CHOICE		EnumRange (OFF ON)
out.25.in.58	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.25.in.59	CHOICE		EnumRange (OFF ON)
out.25.in.60	CHOICE		EnumRange (OFF ON)
out.25.in.61	CHOICE		EnumRange (OFF ON)
out.25.in.62	CHOICE		EnumRange (OFF ON)
out.25.in.63	CHOICE		EnumRange (OFF ON)
out.25.in.64	CHOICE		EnumRange (OFF ON)
out.26.in.01	CHOICE		EnumRange (OFF ON)
out.26.in.02	CHOICE		EnumRange (OFF ON)
out.26.in.03	CHOICE		EnumRange (OFF ON)
out.26.in.04	CHOICE		EnumRange (OFF ON)
out.26.in.05	CHOICE		EnumRange (OFF ON)
out.26.in.06	CHOICE		EnumRange (OFF ON)
out.26.in.07	CHOICE		EnumRange (OFF ON)
out.26.in.08	CHOICE		EnumRange (OFF ON)
out.26.in.09	CHOICE		EnumRange (OFF ON)
out.26.in.10	CHOICE		EnumRange (OFF ON)
out.26.in.11	CHOICE		EnumRange (OFF ON)
out.26.in.12	CHOICE		EnumRange (OFF ON)
out.26.in.13	CHOICE		EnumRange (OFF ON)
out.26.in.14	CHOICE		EnumRange (OFF ON)
out.26.in.15	CHOICE		EnumRange (OFF ON)
out.26.in.16	CHOICE		EnumRange (OFF ON)
out.26.in.17	CHOICE		EnumRange (OFF ON)
out.26.in.18	CHOICE		EnumRange (OFF ON)
out.26.in.19	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.26.in.20	CHOICE		EnumRange (OFF ON)
out.26.in.21	CHOICE		EnumRange (OFF ON)
out.26.in.22	CHOICE		EnumRange (OFF ON)
out.26.in.23	CHOICE		EnumRange (OFF ON)
out.26.in.24	CHOICE		EnumRange (OFF ON)
out.26.in.25	CHOICE		EnumRange (OFF ON)
out.26.in.26	CHOICE		EnumRange (OFF ON)
out.26.in.27	CHOICE		EnumRange (OFF ON)
out.26.in.28	CHOICE		EnumRange (OFF ON)
out.26.in.29	CHOICE		EnumRange (OFF ON)
out.26.in.30	CHOICE		EnumRange (OFF ON)
out.26.in.31	CHOICE		EnumRange (OFF ON)
out.26.in.32	CHOICE		EnumRange (OFF ON)
out.26.in.33	CHOICE		EnumRange (OFF ON)
out.26.in.34	CHOICE		EnumRange (OFF ON)
out.26.in.35	CHOICE		EnumRange (OFF ON)
out.26.in.36	CHOICE		EnumRange (OFF ON)
out.26.in.37	CHOICE		EnumRange (OFF ON)
out.26.in.38	CHOICE		EnumRange (OFF ON)
out.26.in.39	CHOICE		EnumRange (OFF ON)
out.26.in.40	CHOICE		EnumRange (OFF ON)
out.26.in.41	CHOICE		EnumRange (OFF ON)
out.26.in.42	CHOICE		EnumRange (OFF ON)
out.26.in.43	CHOICE		EnumRange (OFF ON)
out.26.in.44	CHOICE		EnumRange (OFF ON)
out.26.in.45	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.26.in.46	CHOICE		EnumRange (OFF ON)
out.26.in.47	CHOICE		EnumRange (OFF ON)
out.26.in.48	CHOICE		EnumRange (OFF ON)
out.26.in.49	CHOICE		EnumRange (OFF ON)
out.26.in.50	CHOICE		EnumRange (OFF ON)
out.26.in.51	CHOICE		EnumRange (OFF ON)
out.26.in.52	CHOICE		EnumRange (OFF ON)
out.26.in.53	CHOICE		EnumRange (OFF ON)
out.26.in.54	CHOICE		EnumRange (OFF ON)
out.26.in.55	CHOICE		EnumRange (OFF ON)
out.26.in.56	CHOICE		EnumRange (OFF ON)
out.26.in.57	CHOICE		EnumRange (OFF ON)
out.26.in.58	CHOICE		EnumRange (OFF ON)
out.26.in.59	CHOICE		EnumRange (OFF ON)
out.26.in.60	CHOICE		EnumRange (OFF ON)
out.26.in.61	CHOICE		EnumRange (OFF ON)
out.26.in.62	CHOICE		EnumRange (OFF ON)
out.26.in.63	CHOICE		EnumRange (OFF ON)
out.26.in.64	CHOICE		EnumRange (OFF ON)
out.27.in.01	CHOICE		EnumRange (OFF ON)
out.27.in.02	CHOICE		EnumRange (OFF ON)
out.27.in.03	CHOICE		EnumRange (OFF ON)
out.27.in.04	CHOICE		EnumRange (OFF ON)
out.27.in.05	CHOICE		EnumRange (OFF ON)
out.27.in.06	CHOICE		EnumRange (OFF ON)
out.27.in.07	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.27.in.08	CHOICE		EnumRange (OFF ON)
out.27.in.09	CHOICE		EnumRange (OFF ON)
out.27.in.10	CHOICE		EnumRange (OFF ON)
out.27.in.11	CHOICE		EnumRange (OFF ON)
out.27.in.12	CHOICE		EnumRange (OFF ON)
out.27.in.13	CHOICE		EnumRange (OFF ON)
out.27.in.14	CHOICE		EnumRange (OFF ON)
out.27.in.15	CHOICE		EnumRange (OFF ON)
out.27.in.16	CHOICE		EnumRange (OFF ON)
out.27.in.17	CHOICE		EnumRange (OFF ON)
out.27.in.18	CHOICE		EnumRange (OFF ON)
out.27.in.19	CHOICE		EnumRange (OFF ON)
out.27.in.20	CHOICE		EnumRange (OFF ON)
out.27.in.21	CHOICE		EnumRange (OFF ON)
out.27.in.22	CHOICE		EnumRange (OFF ON)
out.27.in.23	CHOICE		EnumRange (OFF ON)
out.27.in.24	CHOICE		EnumRange (OFF ON)
out.27.in.25	CHOICE		EnumRange (OFF ON)
out.27.in.26	CHOICE		EnumRange (OFF ON)
out.27.in.27	CHOICE		EnumRange (OFF ON)
out.27.in.28	CHOICE		EnumRange (OFF ON)
out.27.in.29	CHOICE		EnumRange (OFF ON)
out.27.in.30	CHOICE		EnumRange (OFF ON)
out.27.in.31	CHOICE		EnumRange (OFF ON)
out.27.in.32	CHOICE		EnumRange (OFF ON)
out.27.in.33	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.27.in.34	CHOICE		EnumRange (OFF ON)
out.27.in.35	CHOICE		EnumRange (OFF ON)
out.27.in.36	CHOICE		EnumRange (OFF ON)
out.27.in.37	CHOICE		EnumRange (OFF ON)
out.27.in.38	CHOICE		EnumRange (OFF ON)
out.27.in.39	CHOICE		EnumRange (OFF ON)
out.27.in.40	CHOICE		EnumRange (OFF ON)
out.27.in.41	CHOICE		EnumRange (OFF ON)
out.27.in.42	CHOICE		EnumRange (OFF ON)
out.27.in.43	CHOICE		EnumRange (OFF ON)
out.27.in.44	CHOICE		EnumRange (OFF ON)
out.27.in.45	CHOICE		EnumRange (OFF ON)
out.27.in.46	CHOICE		EnumRange (OFF ON)
out.27.in.47	CHOICE		EnumRange (OFF ON)
out.27.in.48	CHOICE		EnumRange (OFF ON)
out.27.in.49	CHOICE		EnumRange (OFF ON)
out.27.in.50	CHOICE		EnumRange (OFF ON)
out.27.in.51	CHOICE		EnumRange (OFF ON)
out.27.in.52	CHOICE		EnumRange (OFF ON)
out.27.in.53	CHOICE		EnumRange (OFF ON)
out.27.in.54	CHOICE		EnumRange (OFF ON)
out.27.in.55	CHOICE		EnumRange (OFF ON)
out.27.in.56	CHOICE		EnumRange (OFF ON)
out.27.in.57	CHOICE		EnumRange (OFF ON)
out.27.in.58	CHOICE		EnumRange (OFF ON)
out.27.in.59	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.27.in.60	CHOICE		EnumRange (OFF ON)
out.27.in.61	CHOICE		EnumRange (OFF ON)
out.27.in.62	CHOICE		EnumRange (OFF ON)
out.27.in.63	CHOICE		EnumRange (OFF ON)
out.27.in.64	CHOICE		EnumRange (OFF ON)
out.28.in.01	CHOICE		EnumRange (OFF ON)
out.28.in.02	CHOICE		EnumRange (OFF ON)
out.28.in.03	CHOICE		EnumRange (OFF ON)
out.28.in.04	CHOICE		EnumRange (OFF ON)
out.28.in.05	CHOICE		EnumRange (OFF ON)
out.28.in.06	CHOICE		EnumRange (OFF ON)
out.28.in.07	CHOICE		EnumRange (OFF ON)
out.28.in.08	CHOICE		EnumRange (OFF ON)
out.28.in.09	CHOICE		EnumRange (OFF ON)
out.28.in.10	CHOICE		EnumRange (OFF ON)
out.28.in.11	CHOICE		EnumRange (OFF ON)
out.28.in.12	CHOICE		EnumRange (OFF ON)
out.28.in.13	CHOICE		EnumRange (OFF ON)
out.28.in.14	CHOICE		EnumRange (OFF ON)
out.28.in.15	CHOICE		EnumRange (OFF ON)
out.28.in.16	CHOICE		EnumRange (OFF ON)
out.28.in.17	CHOICE		EnumRange (OFF ON)
out.28.in.18	CHOICE		EnumRange (OFF ON)
out.28.in.19	CHOICE		EnumRange (OFF ON)
out.28.in.20	CHOICE		EnumRange (OFF ON)
out.28.in.21	CHOICE		EnumRange (OFF ON)
out.28.in.22	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.28.in.23	CHOICE		EnumRange (OFF ON)
out.28.in.24	CHOICE		EnumRange (OFF ON)
out.28.in.25	CHOICE		EnumRange (OFF ON)
out.28.in.26	CHOICE		EnumRange (OFF ON)
out.28.in.27	CHOICE		EnumRange (OFF ON)
out.28.in.28	CHOICE		EnumRange (OFF ON)
out.28.in.29	CHOICE		EnumRange (OFF ON)
out.28.in.30	CHOICE		EnumRange (OFF ON)
out.28.in.31	CHOICE		EnumRange (OFF ON)
out.28.in.32	CHOICE		EnumRange (OFF ON)
out.28.in.33	CHOICE		EnumRange (OFF ON)
out.28.in.34	CHOICE		EnumRange (OFF ON)
out.28.in.35	CHOICE		EnumRange (OFF ON)
out.28.in.36	CHOICE		EnumRange (OFF ON)
out.28.in.37	CHOICE		EnumRange (OFF ON)
out.28.in.38	CHOICE		EnumRange (OFF ON)
out.28.in.39	CHOICE		EnumRange (OFF ON)
out.28.in.40	CHOICE		EnumRange (OFF ON)
out.28.in.41	CHOICE		EnumRange (OFF ON)
out.28.in.42	CHOICE		EnumRange (OFF ON)
out.28.in.43	CHOICE		EnumRange (OFF ON)
out.28.in.44	CHOICE		EnumRange (OFF ON)
out.28.in.45	CHOICE		EnumRange (OFF ON)
out.28.in.46	CHOICE		EnumRange (OFF ON)
out.28.in.47	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.28.in.48	CHOICE		EnumRange (OFF ON)
out.28.in.49	CHOICE		EnumRange (OFF ON)
out.28.in.50	CHOICE		EnumRange (OFF ON)
out.28.in.51	CHOICE		EnumRange (OFF ON)
out.28.in.52	CHOICE		EnumRange (OFF ON)
out.28.in.53	CHOICE		EnumRange (OFF ON)
out.28.in.54	CHOICE		EnumRange (OFF ON)
out.28.in.55	CHOICE		EnumRange (OFF ON)
out.28.in.56	CHOICE		EnumRange (OFF ON)
out.28.in.57	CHOICE		EnumRange (OFF ON)
out.28.in.58	CHOICE		EnumRange (OFF ON)
out.28.in.59	CHOICE		EnumRange (OFF ON)
out.28.in.60	CHOICE		EnumRange (OFF ON)
out.28.in.61	CHOICE		EnumRange (OFF ON)
out.28.in.62	CHOICE		EnumRange (OFF ON)
out.28.in.63	CHOICE		EnumRange (OFF ON)
out.28.in.64	CHOICE		EnumRange (OFF ON)
out.29.in.01	CHOICE		EnumRange (OFF ON)
out.29.in.02	CHOICE		EnumRange (OFF ON)
out.29.in.03	CHOICE		EnumRange (OFF ON)
out.29.in.04	CHOICE		EnumRange (OFF ON)
out.29.in.05	CHOICE		EnumRange (OFF ON)
out.29.in.06	CHOICE		EnumRange (OFF ON)
out.29.in.07	CHOICE		EnumRange (OFF ON)
out.29.in.08	CHOICE		EnumRange (OFF ON)
out.29.in.09	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.29.in.10	CHOICE		EnumRange (OFF ON)
out.29.in.11	CHOICE		EnumRange (OFF ON)
out.29.in.12	CHOICE		EnumRange (OFF ON)
out.29.in.13	CHOICE		EnumRange (OFF ON)
out.29.in.14	CHOICE		EnumRange (OFF ON)
out.29.in.15	CHOICE		EnumRange (OFF ON)
out.29.in.16	CHOICE		EnumRange (OFF ON)
out.29.in.17	CHOICE		EnumRange (OFF ON)
out.29.in.18	CHOICE		EnumRange (OFF ON)
out.29.in.19	CHOICE		EnumRange (OFF ON)
out.29.in.20	CHOICE		EnumRange (OFF ON)
out.29.in.21	CHOICE		EnumRange (OFF ON)
out.29.in.22	CHOICE		EnumRange (OFF ON)
out.29.in.23	CHOICE		EnumRange (OFF ON)
out.29.in.24	CHOICE		EnumRange (OFF ON)
out.29.in.25	CHOICE		EnumRange (OFF ON)
out.29.in.26	CHOICE		EnumRange (OFF ON)
out.29.in.27	CHOICE		EnumRange (OFF ON)
out.29.in.28	CHOICE		EnumRange (OFF ON)
out.29.in.29	CHOICE		EnumRange (OFF ON)
out.29.in.30	CHOICE		EnumRange (OFF ON)
out.29.in.31	CHOICE		EnumRange (OFF ON)
out.29.in.32	CHOICE		EnumRange (OFF ON)
out.29.in.33	CHOICE		EnumRange (OFF ON)
out.29.in.34	CHOICE		EnumRange (OFF ON)
out.29.in.35	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.29.in.36	CHOICE		EnumRange (OFF ON)
out.29.in.37	CHOICE		EnumRange (OFF ON)
out.29.in.38	CHOICE		EnumRange (OFF ON)
out.29.in.39	CHOICE		EnumRange (OFF ON)
out.29.in.40	CHOICE		EnumRange (OFF ON)
out.29.in.41	CHOICE		EnumRange (OFF ON)
out.29.in.42	CHOICE		EnumRange (OFF ON)
out.29.in.43	CHOICE		EnumRange (OFF ON)
out.29.in.44	CHOICE		EnumRange (OFF ON)
out.29.in.45	CHOICE		EnumRange (OFF ON)
out.29.in.46	CHOICE		EnumRange (OFF ON)
out.29.in.47	CHOICE		EnumRange (OFF ON)
out.29.in.48	CHOICE		EnumRange (OFF ON)
out.29.in.49	CHOICE		EnumRange (OFF ON)
out.29.in.50	CHOICE		EnumRange (OFF ON)
out.29.in.51	CHOICE		EnumRange (OFF ON)
out.29.in.52	CHOICE		EnumRange (OFF ON)
out.29.in.53	CHOICE		EnumRange (OFF ON)
out.29.in.54	CHOICE		EnumRange (OFF ON)
out.29.in.55	CHOICE		EnumRange (OFF ON)
out.29.in.56	CHOICE		EnumRange (OFF ON)
out.29.in.57	CHOICE		EnumRange (OFF ON)
out.29.in.58	CHOICE		EnumRange (OFF ON)
out.29.in.59	CHOICE		EnumRange (OFF ON)
out.29.in.60	CHOICE		EnumRange (OFF ON)
out.29.in.61	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.29.in.62	CHOICE		EnumRange (OFF ON)
out.29.in.63	CHOICE		EnumRange (OFF ON)
out.29.in.64	CHOICE		EnumRange (OFF ON)
out.30.in.01	CHOICE		EnumRange (OFF ON)
out.30.in.02	CHOICE		EnumRange (OFF ON)
out.30.in.03	CHOICE		EnumRange (OFF ON)
out.30.in.04	CHOICE		EnumRange (OFF ON)
out.30.in.05	CHOICE		EnumRange (OFF ON)
out.30.in.06	CHOICE		EnumRange (OFF ON)
out.30.in.07	CHOICE		EnumRange (OFF ON)
out.30.in.08	CHOICE		EnumRange (OFF ON)
out.30.in.09	CHOICE		EnumRange (OFF ON)
out.30.in.10	CHOICE		EnumRange (OFF ON)
out.30.in.11	CHOICE		EnumRange (OFF ON)
out.30.in.12	CHOICE		EnumRange (OFF ON)
out.30.in.13	CHOICE		EnumRange (OFF ON)
out.30.in.14	CHOICE		EnumRange (OFF ON)
out.30.in.15	CHOICE		EnumRange (OFF ON)
out.30.in.16	CHOICE		EnumRange (OFF ON)
out.30.in.17	CHOICE		EnumRange (OFF ON)
out.30.in.18	CHOICE		EnumRange (OFF ON)
out.30.in.19	CHOICE		EnumRange (OFF ON)
out.30.in.20	CHOICE		EnumRange (OFF ON)
out.30.in.21	CHOICE		EnumRange (OFF ON)
out.30.in.22	CHOICE		EnumRange (OFF ON)
out.30.in.23	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.30.in.24	CHOICE		EnumRange (OFF ON)
out.30.in.25	CHOICE		EnumRange (OFF ON)
out.30.in.26	CHOICE		EnumRange (OFF ON)
out.30.in.27	CHOICE		EnumRange (OFF ON)
out.30.in.28	CHOICE		EnumRange (OFF ON)
out.30.in.29	CHOICE		EnumRange (OFF ON)
out.30.in.30	CHOICE		EnumRange (OFF ON)
out.30.in.31	CHOICE		EnumRange (OFF ON)
out.30.in.32	CHOICE		EnumRange (OFF ON)
out.30.in.33	CHOICE		EnumRange (OFF ON)
out.30.in.34	CHOICE		EnumRange (OFF ON)
out.30.in.35	CHOICE		EnumRange (OFF ON)
out.30.in.36	CHOICE		EnumRange (OFF ON)
out.30.in.37	CHOICE		EnumRange (OFF ON)
out.30.in.38	CHOICE		EnumRange (OFF ON)
out.30.in.39	CHOICE		EnumRange (OFF ON)
out.30.in.40	CHOICE		EnumRange (OFF ON)
out.30.in.41	CHOICE		EnumRange (OFF ON)
out.30.in.42	CHOICE		EnumRange (OFF ON)
out.30.in.43	CHOICE		EnumRange (OFF ON)
out.30.in.44	CHOICE		EnumRange (OFF ON)
out.30.in.45	CHOICE		EnumRange (OFF ON)
out.30.in.46	CHOICE		EnumRange (OFF ON)
out.30.in.47	CHOICE		EnumRange (OFF ON)
out.30.in.48	CHOICE		EnumRange (OFF ON)
out.30.in.49	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.30.in.50	CHOICE		EnumRange (OFF ON)
out.30.in.51	CHOICE		EnumRange (OFF ON)
out.30.in.52	CHOICE		EnumRange (OFF ON)
out.30.in.53	CHOICE		EnumRange (OFF ON)
out.30.in.54	CHOICE		EnumRange (OFF ON)
out.30.in.55	CHOICE		EnumRange (OFF ON)
out.30.in.56	CHOICE		EnumRange (OFF ON)
out.30.in.57	CHOICE		EnumRange (OFF ON)
out.30.in.58	CHOICE		EnumRange (OFF ON)
out.30.in.59	CHOICE		EnumRange (OFF ON)
out.30.in.60	CHOICE		EnumRange (OFF ON)
out.30.in.61	CHOICE		EnumRange (OFF ON)
out.30.in.62	CHOICE		EnumRange (OFF ON)
out.30.in.63	CHOICE		EnumRange (OFF ON)
out.30.in.64	CHOICE		EnumRange (OFF ON)
out.31.in.01	CHOICE		EnumRange (OFF ON)
out.31.in.02	CHOICE		EnumRange (OFF ON)
out.31.in.03	CHOICE		EnumRange (OFF ON)
out.31.in.04	CHOICE		EnumRange (OFF ON)
out.31.in.05	CHOICE		EnumRange (OFF ON)
out.31.in.06	CHOICE		EnumRange (OFF ON)
out.31.in.07	CHOICE		EnumRange (OFF ON)
out.31.in.08	CHOICE		EnumRange (OFF ON)
out.31.in.09	CHOICE		EnumRange (OFF ON)
out.31.in.10	CHOICE		EnumRange (OFF ON)
out.31.in.11	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.31.in.12	CHOICE		EnumRange (OFF ON)
out.31.in.13	CHOICE		EnumRange (OFF ON)
out.31.in.14	CHOICE		EnumRange (OFF ON)
out.31.in.15	CHOICE		EnumRange (OFF ON)
out.31.in.16	CHOICE		EnumRange (OFF ON)
out.31.in.17	CHOICE		EnumRange (OFF ON)
out.31.in.18	CHOICE		EnumRange (OFF ON)
out.31.in.19	CHOICE		EnumRange (OFF ON)
out.31.in.20	CHOICE		EnumRange (OFF ON)
out.31.in.21	CHOICE		EnumRange (OFF ON)
out.31.in.22	CHOICE		EnumRange (OFF ON)
out.31.in.23	CHOICE		EnumRange (OFF ON)
out.31.in.24	CHOICE		EnumRange (OFF ON)
out.31.in.25	CHOICE		EnumRange (OFF ON)
out.31.in.26	CHOICE		EnumRange (OFF ON)
out.31.in.27	CHOICE		EnumRange (OFF ON)
out.31.in.28	CHOICE		EnumRange (OFF ON)
out.31.in.29	CHOICE		EnumRange (OFF ON)
out.31.in.30	CHOICE		EnumRange (OFF ON)
out.31.in.31	CHOICE		EnumRange (OFF ON)
out.31.in.32	CHOICE		EnumRange (OFF ON)
out.31.in.33	CHOICE		EnumRange (OFF ON)
out.31.in.34	CHOICE		EnumRange (OFF ON)
out.31.in.35	CHOICE		EnumRange (OFF ON)
out.31.in.36	CHOICE		EnumRange (OFF ON)
out.31.in.37	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.31.in.38	CHOICE		EnumRange (OFF ON)
out.31.in.39	CHOICE		EnumRange (OFF ON)
out.31.in.40	CHOICE		EnumRange (OFF ON)
out.31.in.41	CHOICE		EnumRange (OFF ON)
out.31.in.42	CHOICE		EnumRange (OFF ON)
out.31.in.43	CHOICE		EnumRange (OFF ON)
out.31.in.44	CHOICE		EnumRange (OFF ON)
out.31.in.45	CHOICE		EnumRange (OFF ON)
out.31.in.46	CHOICE		EnumRange (OFF ON)
out.31.in.47	CHOICE		EnumRange (OFF ON)
out.31.in.48	CHOICE		EnumRange (OFF ON)
out.31.in.49	CHOICE		EnumRange (OFF ON)
out.31.in.50	CHOICE		EnumRange (OFF ON)
out.31.in.51	CHOICE		EnumRange (OFF ON)
out.31.in.52	CHOICE		EnumRange (OFF ON)
out.31.in.53	CHOICE		EnumRange (OFF ON)
out.31.in.54	CHOICE		EnumRange (OFF ON)
out.31.in.55	CHOICE		EnumRange (OFF ON)
out.31.in.56	CHOICE		EnumRange (OFF ON)
out.31.in.57	CHOICE		EnumRange (OFF ON)
out.31.in.58	CHOICE		EnumRange (OFF ON)
out.31.in.59	CHOICE		EnumRange (OFF ON)
out.31.in.60	CHOICE		EnumRange (OFF ON)
out.31.in.61	CHOICE		EnumRange (OFF ON)
out.31.in.62	CHOICE		EnumRange (OFF ON)
out.31.in.63	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.31.in.64	CHOICE		EnumRange (OFF ON)
out.32.in.01	CHOICE		EnumRange (OFF ON)
out.32.in.02	CHOICE		EnumRange (OFF ON)
out.32.in.03	CHOICE		EnumRange (OFF ON)
out.32.in.04	CHOICE		EnumRange (OFF ON)
out.32.in.05	CHOICE		EnumRange (OFF ON)
out.32.in.06	CHOICE		EnumRange (OFF ON)
out.32.in.07	CHOICE		EnumRange (OFF ON)
out.32.in.08	CHOICE		EnumRange (OFF ON)
out.32.in.09	CHOICE		EnumRange (OFF ON)
out.32.in.10	CHOICE		EnumRange (OFF ON)
out.32.in.11	CHOICE		EnumRange (OFF ON)
out.32.in.12	CHOICE		EnumRange (OFF ON)
out.32.in.13	CHOICE		EnumRange (OFF ON)
out.32.in.14	CHOICE		EnumRange (OFF ON)
out.32.in.15	CHOICE		EnumRange (OFF ON)
out.32.in.16	CHOICE		EnumRange (OFF ON)
out.32.in.17	CHOICE		EnumRange (OFF ON)
out.32.in.18	CHOICE		EnumRange (OFF ON)
out.32.in.19	CHOICE		EnumRange (OFF ON)
out.32.in.20	CHOICE		EnumRange (OFF ON)
out.32.in.21	CHOICE		EnumRange (OFF ON)
out.32.in.22	CHOICE		EnumRange (OFF ON)
out.32.in.23	CHOICE		EnumRange (OFF ON)
out.32.in.24	CHOICE		EnumRange (OFF ON)
out.32.in.25	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.32.in.26	CHOICE		EnumRange (OFF ON)
out.32.in.27	CHOICE		EnumRange (OFF ON)
out.32.in.28	CHOICE		EnumRange (OFF ON)
out.32.in.29	CHOICE		EnumRange (OFF ON)
out.32.in.30	CHOICE		EnumRange (OFF ON)
out.32.in.31	CHOICE		EnumRange (OFF ON)
out.32.in.32	CHOICE		EnumRange (OFF ON)
out.32.in.33	CHOICE		EnumRange (OFF ON)
out.32.in.34	CHOICE		EnumRange (OFF ON)
out.32.in.35	CHOICE		EnumRange (OFF ON)
out.32.in.36	CHOICE		EnumRange (OFF ON)
out.32.in.37	CHOICE		EnumRange (OFF ON)
out.32.in.38	CHOICE		EnumRange (OFF ON)
out.32.in.39	CHOICE		EnumRange (OFF ON)
out.32.in.40	CHOICE		EnumRange (OFF ON)
out.32.in.41	CHOICE		EnumRange (OFF ON)
out.32.in.42	CHOICE		EnumRange (OFF ON)
out.32.in.43	CHOICE		EnumRange (OFF ON)
out.32.in.44	CHOICE		EnumRange (OFF ON)
out.32.in.45	CHOICE		EnumRange (OFF ON)
out.32.in.46	CHOICE		EnumRange (OFF ON)
out.32.in.47	CHOICE		EnumRange (OFF ON)
out.32.in.48	CHOICE		EnumRange (OFF ON)
out.32.in.49	CHOICE		EnumRange (OFF ON)
out.32.in.50	CHOICE		EnumRange (OFF ON)
out.32.in.51	CHOICE		EnumRange (OFF ON)
out.32.in.52	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.32.in.53	CHOICE		EnumRange (OFF ON)
out.32.in.54	CHOICE		EnumRange (OFF ON)
out.32.in.55	CHOICE		EnumRange (OFF ON)
out.32.in.56	CHOICE		EnumRange (OFF ON)
out.32.in.57	CHOICE		EnumRange (OFF ON)
out.32.in.58	CHOICE		EnumRange (OFF ON)
out.32.in.59	CHOICE		EnumRange (OFF ON)
out.32.in.60	CHOICE		EnumRange (OFF ON)
out.32.in.61	CHOICE		EnumRange (OFF ON)
out.32.in.62	CHOICE		EnumRange (OFF ON)
out.32.in.63	CHOICE		EnumRange (OFF ON)
out.32.in.64	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan PSU2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temp PSU1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temp PSU2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Backup Active)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 PSU2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Fan PSU2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp PSU1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Temp PSU2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Monitoring)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Chassis 2 Monitoring)
faults.101	ALARM	R/O	AlarmFlagRange R/O (Input 1 Summary)
faults.102	ALARM	R/O	AlarmFlagRange R/O (Input 2 Summary)
faults.103	ALARM	R/O	AlarmFlagRange R/O (Input 3 Summary)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Input 4 Summary)
faults.105	ALARM	R/O	AlarmFlagRange R/O (Input 5 Summary)
faults.106	ALARM	R/O	AlarmFlagRange R/O (Input 6 Summary)
faults.107	ALARM	R/O	AlarmFlagRange R/O (Input 7 Summary)

name	*type*	*flags*	*range*
faults.108	ALARM	R/O	AlarmFlagRange R/O (Input 8 Summary)
faults.109	ALARM	R/O	AlarmFlagRange R/O (Input 9 Summary)
faults.110	ALARM	R/O	AlarmFlagRange R/O (Input 10 Summary)
faults.111	ALARM	R/O	AlarmFlagRange R/O (Input 11 Summary)
faults.112	ALARM	R/O	AlarmFlagRange R/O (Input 12 Summary)
faults.113	ALARM	R/O	AlarmFlagRange R/O (Input 13 Summary)
faults.114	ALARM	R/O	AlarmFlagRange R/O (Input 14 Summary)
faults.115	ALARM	R/O	AlarmFlagRange R/O (Input 15 Summary)
faults.116	ALARM	R/O	AlarmFlagRange R/O (Input 16 Summary)
faults.117	ALARM	R/O	AlarmFlagRange R/O (Input 17 Summary)
faults.118	ALARM	R/O	AlarmFlagRange R/O (Input 18 Summary)
faults.119	ALARM	R/O	AlarmFlagRange R/O (Input 19 Summary)
faults.120	ALARM	R/O	AlarmFlagRange R/O (Input 20 Summary)
faults.121	ALARM	R/O	AlarmFlagRange R/O (Input 21 Summary)
faults.122	ALARM	R/O	AlarmFlagRange R/O (Input 22 Summary)
faults.123	ALARM	R/O	AlarmFlagRange R/O (Input 23 Summary)

name	*type*	*flags*	*range*
faults.124	ALARM	R/O	AlarmFlagRange R/O (Input 24 Summary)
faults.125	ALARM	R/O	AlarmFlagRange R/O (Input 25 Summary)
faults.126	ALARM	R/O	AlarmFlagRange R/O (Input 26 Summary)
faults.127	ALARM	R/O	AlarmFlagRange R/O (Input 27 Summary)
faults.128	ALARM	R/O	AlarmFlagRange R/O (Input 28 Summary)
faults.129	ALARM	R/O	AlarmFlagRange R/O (Input 29 Summary)
faults.130	ALARM	R/O	AlarmFlagRange R/O (Input 30 Summary)
faults.131	ALARM	R/O	AlarmFlagRange R/O (Input 31 Summary)
faults.132	ALARM	R/O	AlarmFlagRange R/O (Input 32 Summary)
faults.133	ALARM	R/O	AlarmFlagRange R/O (Input 33 Summary)
faults.134	ALARM	R/O	AlarmFlagRange R/O (Input 34 Summary)
faults.135	ALARM	R/O	AlarmFlagRange R/O (Input 35 Summary)
faults.136	ALARM	R/O	AlarmFlagRange R/O (Input 36 Summary)
faults.137	ALARM	R/O	AlarmFlagRange R/O (Input 37 Summary)
faults.138	ALARM	R/O	AlarmFlagRange R/O (Input 38 Summary)
faults.139	ALARM	R/O	AlarmFlagRange R/O (Input 39 Summary)

name	*type*	*flags*	*range*
faults.140	ALARM	R/O	AlarmFlagRange R/O (Input 40 Summary)
faults.141	ALARM	R/O	AlarmFlagRange R/O (Input 41 Summary)
faults.142	ALARM	R/O	AlarmFlagRange R/O (Input 42 Summary)
faults.143	ALARM	R/O	AlarmFlagRange R/O (Input 43 Summary)
faults.144	ALARM	R/O	AlarmFlagRange R/O (Input 44 Summary)
faults.145	ALARM	R/O	AlarmFlagRange R/O (Input 45 Summary)
faults.146	ALARM	R/O	AlarmFlagRange R/O (Input 46 Summary)
faults.147	ALARM	R/O	AlarmFlagRange R/O (Input 47 Summary)
faults.148	ALARM	R/O	AlarmFlagRange R/O (Input 48 Summary)
faults.149	ALARM	R/O	AlarmFlagRange R/O (Input 49 Summary)
faults.150	ALARM	R/O	AlarmFlagRange R/O (Input 50 Summary)
faults.151	ALARM	R/O	AlarmFlagRange R/O (Input 51 Summary)
faults.152	ALARM	R/O	AlarmFlagRange R/O (Input 52 Summary)
faults.153	ALARM	R/O	AlarmFlagRange R/O (Input 53 Summary)
faults.154	ALARM	R/O	AlarmFlagRange R/O (Input 54 Summary)
faults.155	ALARM	R/O	AlarmFlagRange R/O (Input 55 Summary)

name	*type*	*flags*	*range*
faults.156	ALARM	R/O	AlarmFlagRange R/O (Input 56 Summary)
faults.157	ALARM	R/O	AlarmFlagRange R/O (Input 57 Summary)
faults.158	ALARM	R/O	AlarmFlagRange R/O (Input 58 Summary)
faults.159	ALARM	R/O	AlarmFlagRange R/O (Input 59 Summary)
faults.160	ALARM	R/O	AlarmFlagRange R/O (Input 60 Summary)
faults.161	ALARM	R/O	AlarmFlagRange R/O (Input 61 Summary)
faults.162	ALARM	R/O	AlarmFlagRange R/O (Input 62 Summary)
faults.163	ALARM	R/O	AlarmFlagRange R/O (Input 63 Summary)
faults.164	ALARM	R/O	AlarmFlagRange R/O (Input 64 Summary)
faults.201	ALARM	R/O	AlarmFlagRange R/O (Output 1 Summary)
faults.202	ALARM	R/O	AlarmFlagRange R/O (Output 2 Summary)
faults.203	ALARM	R/O	AlarmFlagRange R/O (Output 3 Summary)
faults.204	ALARM	R/O	AlarmFlagRange R/O (Output 4 Summary)
faults.205	ALARM	R/O	AlarmFlagRange R/O (Output 5 Summary)
faults.206	ALARM	R/O	AlarmFlagRange R/O (Output 6 Summary)
faults.207	ALARM	R/O	AlarmFlagRange R/O (Output 7 Summary)

name	*type*	*flags*	*range*
faults.208	ALARM	R/O	AlarmFlagRange R/O (Output 8 Summary)
faults.209	ALARM	R/O	AlarmFlagRange R/O (Output 9 Summary)
faults.210	ALARM	R/O	AlarmFlagRange R/O (Output 10 Summary)
faults.211	ALARM	R/O	AlarmFlagRange R/O (Output 11 Summary)
faults.212	ALARM	R/O	AlarmFlagRange R/O (Output 12 Summary)
faults.213	ALARM	R/O	AlarmFlagRange R/O (Output 13 Summary)
faults.214	ALARM	R/O	AlarmFlagRange R/O (Output 14 Summary)
faults.215	ALARM	R/O	AlarmFlagRange R/O (Output 15 Summary)
faults.216	ALARM	R/O	AlarmFlagRange R/O (Output 16 Summary)
faults.217	ALARM	R/O	AlarmFlagRange R/O (Output 17 Summary)
faults.218	ALARM	R/O	AlarmFlagRange R/O (Output 18 Summary)
faults.219	ALARM	R/O	AlarmFlagRange R/O (Output 19 Summary)
faults.220	ALARM	R/O	AlarmFlagRange R/O (Output 20 Summary)
faults.221	ALARM	R/O	AlarmFlagRange R/O (Output 21 Summary)
faults.222	ALARM	R/O	AlarmFlagRange R/O (Output 22 Summary)
faults.223	ALARM	R/O	AlarmFlagRange R/O (Output 23 Summary)

name	*type*	*flags*	*range*
faults.224	ALARM	R/O	AlarmFlagRange R/O (Output 24 Summary)
faults.225	ALARM	R/O	AlarmFlagRange R/O (Output 25 Summary)
faults.226	ALARM	R/O	AlarmFlagRange R/O (Output 26 Summary)
faults.227	ALARM	R/O	AlarmFlagRange R/O (Output 27 Summary)
faults.228	ALARM	R/O	AlarmFlagRange R/O (Output 28 Summary)
faults.229	ALARM	R/O	AlarmFlagRange R/O (Output 29 Summary)
faults.230	ALARM	R/O	AlarmFlagRange R/O (Output 30 Summary)
faults.231	ALARM	R/O	AlarmFlagRange R/O (Output 31 Summary)
faults.232	ALARM	R/O	AlarmFlagRange R/O (Output 32 Summary)
faults.233	ALARM	R/O	AlarmFlagRange R/O (Output 33 Summary)
faults.234	ALARM	R/O	AlarmFlagRange R/O (Output 34 Summary)
faults.235	ALARM	R/O	AlarmFlagRange R/O (Output 35 Summary)
faults.236	ALARM	R/O	AlarmFlagRange R/O (Output 36 Summary)
faults.237	ALARM	R/O	AlarmFlagRange R/O (Output 37 Summary)
faults.238	ALARM	R/O	AlarmFlagRange R/O (Output 38 Summary)
faults.239	ALARM	R/O	AlarmFlagRange R/O (Output 39 Summary)

name	*type*	*flags*	*range*
faults.240	ALARM	R/O	AlarmFlagRange R/O (Output 40 Summary)
faults.241	ALARM	R/O	AlarmFlagRange R/O (Output 41 Summary)
faults.242	ALARM	R/O	AlarmFlagRange R/O (Output 42 Summary)
faults.243	ALARM	R/O	AlarmFlagRange R/O (Output 43 Summary)
faults.244	ALARM	R/O	AlarmFlagRange R/O (Output 44 Summary)
faults.245	ALARM	R/O	AlarmFlagRange R/O (Output 45 Summary)
faults.246	ALARM	R/O	AlarmFlagRange R/O (Output 46 Summary)
faults.247	ALARM	R/O	AlarmFlagRange R/O (Output 47 Summary)
faults.248	ALARM	R/O	AlarmFlagRange R/O (Output 48 Summary)
faults.249	ALARM	R/O	AlarmFlagRange R/O (Output 49 Summary)
faults.250	ALARM	R/O	AlarmFlagRange R/O (Output 50 Summary)
faults.251	ALARM	R/O	AlarmFlagRange R/O (Output 51 Summary)
faults.252	ALARM	R/O	AlarmFlagRange R/O (Output 52 Summary)
faults.253	ALARM	R/O	AlarmFlagRange R/O (Output 53 Summary)
faults.254	ALARM	R/O	AlarmFlagRange R/O (Output 54 Summary)
faults.255	ALARM	R/O	AlarmFlagRange R/O (Output 55 Summary)

name	*type*	*flags*	*range*
faults.256	ALARM	R/O	AlarmFlagRange R/O (Output 56 Summary)
faults.257	ALARM	R/O	AlarmFlagRange R/O (Output 57 Summary)
faults.258	ALARM	R/O	AlarmFlagRange R/O (Output 58 Summary)
faults.259	ALARM	R/O	AlarmFlagRange R/O (Output 59 Summary)
faults.260	ALARM	R/O	AlarmFlagRange R/O (Output 60 Summary)
faults.261	ALARM	R/O	AlarmFlagRange R/O (Output 61 Summary)
faults.262	ALARM	R/O	AlarmFlagRange R/O (Output 62 Summary)
faults.263	ALARM	R/O	AlarmFlagRange R/O (Output 63 Summary)
faults.264	ALARM	R/O	AlarmFlagRange R/O (Output 64 Summary)

9.210 ETL-PDSwitch-Emulator

The *ETL-PDSwitch-Emulator* logical device emulates a N:1 RF Switch like the ETL CST-POE-SW02-1059 PIN Diode 2:1 switch. It listens at a configurable IP port and offers there a simplified version of the remote protocol of the emulated device. Switch positions may be set either with the 'position' variable of the driver or with commands sent to the IP port the device is listening to.

The device supports four different operating modes which control up to four IO lines in different configurations:

TYPE-2WAY-1BIT

This mode emulates a 2-way switch which is controlled by one IO line. Valid position values are "01" and "02" in this mode, "02" switches the IO line to "ON", "01" switches it "OFF".

TYPE-2WAY-2BIT

This mode also emulates a 2-way switch, but this is controlled by two IO lines. Valid position values are "00", "01" and "02" in this mode:

switch position	00	01	02
bit / IO-line 01	OFF	ON	OFF
bit / IO-line 02	OFF	OFF	ON

A state where both IO-lines are set "ON" is never commanded by the device, if this happens by external control of the IO-lines, this is treated as a fault.

TYPE-4WAY-2BIT

This mode emulates a 4-way switch which is controlled by two IO lines. Valid position values are "01" to "04" in this mode, the positions are encoded as a binary number:

switch position	01	02	03	04
bit / IO-line 01	OFF	ON	OFF	ON
bit / IO-line 02	OFF	OFF	ON	ON

TYPE-4WAY-4BIT

This mode emulates a 4-way switch which is controlled by four IO lines. Valid position values are "00" to "04" in this mode, the positions are encoded as shown in the table below:

switch position	00	01	02	03	04
bit / IO-line 01	OFF	ON	OFF	OFF	OFF
bit / IO-line 02	OFF	OFF	ON	OFF	OFF
bit / IO-line 03	OFF	OFF	OFF	ON	OFF
bit / IO-line 04	OFF	OFF	OFF	OFF	ON

With this mode there may be at max one IO-line switched on at a time, this line sets the switch position 01..04. If the device detects more than one active IO-lines, this is treated a a fault.

The device unterstands two commands to query the actual status and to set the switch to a certain position:

- Sending `{A?}` to the device queries the actual switch setting. The device responds with `{A,nn}` where *nn* is the number of the actually selected port (2-digit, leading zero).*00* means no port is selected, *01* port number 1 and so on.
- Sending `{ACnn}` to the device commands a new switch setting. The device responds with `{A,nn}` where *nn* is the number of the actually selected port (2-digit, leading zero). *00* means no port is selected,*01* port number 1 and so on.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch position.
- --- The device info page.
- --- Shows the device's faults.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the *ETL-PDSwitch-Emulator* device with the device it shall control.

parameter	***description***
portNo	This configures the port number the device is listening. The valid range is 1024 .. 65535. You cannot use a port number that is already used by another application. The device will show a fault if it cannot open the port because it is already in use.
switchType	Sets one of the operating modes described above. May be set to one of TYPE-2WAY-1BIT, TYPE-2WAY-2BIT, TYPE-4WAY-2BIT or TYPE-4WAY-4BIT
bitSense	May be used to invert the meaning of ON/OFF. If set to NORMAL, the device commands the IO-lines as described above to ON and OFF. If INVERTED is selected ON and OFF are interchanged.
bitId.01 .. bitId.32	These are the message identifiers for the up to 4 IO lines to control. Unused outputs should be left empty.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.bitval.01	TEXT	R/O	StringRange R/O
info.bitval.02	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.bitval.03	TEXT	R/O	StringRange R/O
info.bitval.04	TEXT	R/O	StringRange R/O
info.decodedPos	TEXT	R/O	StringRange R/O
config.portNo	INTEGER	SAVE SETUP	IntegerRange (1024 .. 65535)
config.switchType	CHOICE	SAVE SETUP	EnumRange (TYPE-UNKNOWN TYPE-2WAY-1BIT TYPE-2WAY-2BIT TYPE-4WAY-2BIT TYPE-4WAY-4BIT)
config.bitSense	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERTED)
config.bitId.01	TEXT	SAVE SETUP	StringRange
config.bitId.02	TEXT	SAVE SETUP	StringRange
config.bitId.03	TEXT	SAVE SETUP	StringRange
config.bitId.04	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (00 01 02 03 04)
onValue	TEXT	R/O	StringRange R/O
offValue	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IP Port)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch type)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch position)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Bit combination)

9.211 ETL-POE-Switch

This page describes the *ETL-POE-Switch* device driver and its device window. The *ETL-POE-Switch* device driver monitors and controls the features and alarms of one ETL Power over Ethernet 4:1 or 2:1 switch.

The driver was tested with the following equipment:

- 4-Port Switch
- SW version: e401 1v1

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch position.
- --- The faults page.
- --- The device info page including units temperature and psu voltages.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- inputs --- Selects if the unit is equipped either 2 or 4 inputs.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position	CHOICE		EnumRange (1 2 3 4)
reset	TEXT	NOPRESET	StringRange
internal.choices	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.212 ETL-Piranha

This page describes the SNMP based device driver for the Piranha Series LNA DC power supply from ETL Systems.

The driver detects the number of installed modules and their voltage options automatically.

The driver was tested with the following model:

- Piranha PRN-15 with 16 modules and external reference
- SW release e435 1v05

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The I/O controller parameters and sensor measurements
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` and `private` as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.bootVersion	TEXT	R/O	StringRange R/O
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.12.chars	TEXT	R/O	StringRange R/O
internal.intLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.extGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.voltage	TEXT	R/O	StringRange R/O
internal.voltageOpt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.off	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.cus	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.tra	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.3k3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.5V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.8V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.13V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.15V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.18V	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.20V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.24V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.36V	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.48V	BOOLEAN	R/O	BooleanRange R/O (true false)
ref.mode	CHOICE		EnumRange (INTERNAL EXTERNAL)
ref.int	CHOICE		EnumRange (OFF ON)
ref.extGain	FLOAT		DoubleRange (-10.0 .. 15.0)
ref.intLevel	FLOAT		DoubleRange (-10.0 .. 15.0)
ref.present	TEXT	R/O	StringRange R/O
ref.extLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.type	TEXT	R/O	StringRange R/O
psu.1.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.powerRemaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.hoursElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.fan.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.fan.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.1.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.1.tempState	TEXT	R/O	StringRange R/O
psu.1.presentState	TEXT	R/O	StringRange R/O
psu.2.type	TEXT	R/O	StringRange R/O
psu.2.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.powerRemaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.hoursElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
psu.2.fan.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.fan.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
psu.2.tempState	TEXT	R/O	StringRange R/O
psu.2.presentState	TEXT	R/O	StringRange R/O
info.cpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
module.01.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.01.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.01.22kHz	CHOICE		EnumRange (OFF ON)
module.01.reference	CHOICE		EnumRange (OFF ON)
module.01.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.01.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.01.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.01.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.01.name	TEXT		StringRange
module.01.present	TEXT	R/O	StringRange R/O
module.01.type	TEXT	R/O	StringRange R/O
module.01.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.01.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.01.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.01.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.01.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.01.voltageState	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.01.currentState	TEXT	R/O	StringRange R/O
module.01.rfLevelState	TEXT	R/O	StringRange R/O
module.01.ampVoltageState	TEXT	R/O	StringRange R/O
module.01.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.01.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.01.tempState	TEXT	R/O	StringRange R/O
module.02.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.02.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.02.22kHz	CHOICE		EnumRange (OFF ON)
module.02.reference	CHOICE		EnumRange (OFF ON)
module.02.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.02.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.02.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.02.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.02.name	TEXT		StringRange
module.02.present	TEXT	R/O	StringRange R/O
module.02.type	TEXT	R/O	StringRange R/O
module.02.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.02.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.02.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.02.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.02.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.02.voltageState	TEXT	R/O	StringRange R/O
module.02.currentState	TEXT	R/O	StringRange R/O
module.02.rfLevelState	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.02.ampVoltageState	TEXT	R/O	StringRange R/O
module.02.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.02.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.02.tempState	TEXT	R/O	StringRange R/O
module.03.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.03.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.03.22kHz	CHOICE		EnumRange (OFF ON)
module.03.reference	CHOICE		EnumRange (OFF ON)
module.03.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.03.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.03.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.03.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.03.name	TEXT		StringRange
module.03.present	TEXT	R/O	StringRange R/O
module.03.type	TEXT	R/O	StringRange R/O
module.03.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.03.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.03.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.03.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.03.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.03.voltageState	TEXT	R/O	StringRange R/O
module.03.currentState	TEXT	R/O	StringRange R/O
module.03.rfLevelState	TEXT	R/O	StringRange R/O
module.03.ampVoltageState	TEXT	R/O	StringRange R/O
module.03.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
module.03.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.03.tempState	TEXT	R/O	StringRange R/O
module.04.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.04.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.04.22kHz	CHOICE		EnumRange (OFF ON)
module.04.reference	CHOICE		EnumRange (OFF ON)
module.04.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.04.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.04.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.04.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.04.name	TEXT		StringRange
module.04.present	TEXT	R/O	StringRange R/O
module.04.type	TEXT	R/O	StringRange R/O
module.04.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.04.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.04.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.04.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.04.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.04.voltageState	TEXT	R/O	StringRange R/O
module.04.currentState	TEXT	R/O	StringRange R/O
module.04.rfLevelState	TEXT	R/O	StringRange R/O
module.04.ampVoltageState	TEXT	R/O	StringRange R/O
module.04.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.04.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.04.tempState	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.05.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.05.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.05.22kHz	CHOICE		EnumRange (OFF ON)
module.05.reference	CHOICE		EnumRange (OFF ON)
module.05.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.05.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.05.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.05.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.05.name	TEXT		StringRange
module.05.present	TEXT	R/O	StringRange R/O
module.05.type	TEXT	R/O	StringRange R/O
module.05.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.05.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.05.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.05.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.05.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.05.voltageState	TEXT	R/O	StringRange R/O
module.05.currentState	TEXT	R/O	StringRange R/O
module.05.rfLevelState	TEXT	R/O	StringRange R/O
module.05.ampVoltageState	TEXT	R/O	StringRange R/O
module.05.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.05.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.05.tempState	TEXT	R/O	StringRange R/O
module.06.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)

name	*type*	*flags*	*range*
module.06.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.06.22kHz	CHOICE		EnumRange (OFF ON)
module.06.reference	CHOICE		EnumRange (OFF ON)
module.06.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.06.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.06.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.06.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.06.name	TEXT		StringRange
module.06.present	TEXT	R/O	StringRange R/O
module.06.type	TEXT	R/O	StringRange R/O
module.06.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.06.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.06.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.06.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.06.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.06.voltageState	TEXT	R/O	StringRange R/O
module.06.currentState	TEXT	R/O	StringRange R/O
module.06.rfLevelState	TEXT	R/O	StringRange R/O
module.06.ampVoltageState	TEXT	R/O	StringRange R/O
module.06.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.06.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.06.tempState	TEXT	R/O	StringRange R/O
module.07.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.07.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.07.22kHz	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
module.07.reference	CHOICE		EnumRange (OFF ON)
module.07.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.07.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.07.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.07.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.07.name	TEXT		StringRange
module.07.present	TEXT	R/O	StringRange R/O
module.07.type	TEXT	R/O	StringRange R/O
module.07.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.07.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.07.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.07.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.07.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.07.voltageState	TEXT	R/O	StringRange R/O
module.07.currentState	TEXT	R/O	StringRange R/O
module.07.rfLevelState	TEXT	R/O	StringRange R/O
module.07.ampVoltageState	TEXT	R/O	StringRange R/O
module.07.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.07.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.07.tempState	TEXT	R/O	StringRange R/O
module.08.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.08.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.08.22kHz	CHOICE		EnumRange (OFF ON)
module.08.reference	CHOICE		EnumRange (OFF ON)
module.08.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)

name	*type*	*flags*	*range*
module.08.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.08.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.08.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.08.name	TEXT		StringRange
module.08.present	TEXT	R/O	StringRange R/O
module.08.type	TEXT	R/O	StringRange R/O
module.08.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.08.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.08.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.08.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.08.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.08.voltageState	TEXT	R/O	StringRange R/O
module.08.currentState	TEXT	R/O	StringRange R/O
module.08.rfLevelState	TEXT	R/O	StringRange R/O
module.08.ampVoltageState	TEXT	R/O	StringRange R/O
module.08.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.08.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.08.tempState	TEXT	R/O	StringRange R/O
module.09.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.09.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.09.22kHz	CHOICE		EnumRange (OFF ON)
module.09.reference	CHOICE		EnumRange (OFF ON)
module.09.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.09.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.09.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)

name	*type*	*flags*	*range*
module.09.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.09.name	TEXT		StringRange
module.09.present	TEXT	R/O	StringRange R/O
module.09.type	TEXT	R/O	StringRange R/O
module.09.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.09.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.09.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.09.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.09.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.09.voltageState	TEXT	R/O	StringRange R/O
module.09.currentState	TEXT	R/O	StringRange R/O
module.09.rfLevelState	TEXT	R/O	StringRange R/O
module.09.ampVoltageState	TEXT	R/O	StringRange R/O
module.09.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.09.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.09.tempState	TEXT	R/O	StringRange R/O
module.10.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.10.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.10.22kHz	CHOICE		EnumRange (OFF ON)
module.10.reference	CHOICE		EnumRange (OFF ON)
module.10.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.10.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.10.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.10.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.10.name	TEXT		StringRange

name	*type*	*flags*	*range*
module.10.present	TEXT	R/O	StringRange R/O
module.10.type	TEXT	R/O	StringRange R/O
module.10.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.10.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.10.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.10.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.10.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.10.voltageState	TEXT	R/O	StringRange R/O
module.10.currentState	TEXT	R/O	StringRange R/O
module.10.rfLevelState	TEXT	R/O	StringRange R/O
module.10.ampVoltageState	TEXT	R/O	StringRange R/O
module.10.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.10.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.10.tempState	TEXT	R/O	StringRange R/O
module.11.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.11.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.11.22kHz	CHOICE		EnumRange (OFF ON)
module.11.reference	CHOICE		EnumRange (OFF ON)
module.11.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.11.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.11.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.11.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.11.name	TEXT		StringRange
module.11.present	TEXT	R/O	StringRange R/O
module.11.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.11.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.11.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.11.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.11.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.11.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.11.voltageState	TEXT	R/O	StringRange R/O
module.11.currentState	TEXT	R/O	StringRange R/O
module.11.rfLevelState	TEXT	R/O	StringRange R/O
module.11.ampVoltageState	TEXT	R/O	StringRange R/O
module.11.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.11.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.11.tempState	TEXT	R/O	StringRange R/O
module.12.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.12.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.12.22kHz	CHOICE		EnumRange (OFF ON)
module.12.reference	CHOICE		EnumRange (OFF ON)
module.12.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.12.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.12.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.12.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.12.name	TEXT		StringRange
module.12.present	TEXT	R/O	StringRange R/O
module.12.type	TEXT	R/O	StringRange R/O
module.12.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.12.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
module.12.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.12.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.12.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.12.voltageState	TEXT	R/O	StringRange R/O
module.12.currentState	TEXT	R/O	StringRange R/O
module.12.rfLevelState	TEXT	R/O	StringRange R/O
module.12.ampVoltageState	TEXT	R/O	StringRange R/O
module.12.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.12.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.12.tempState	TEXT	R/O	StringRange R/O
module.13.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.13.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.13.22kHz	CHOICE		EnumRange (OFF ON)
module.13.reference	CHOICE		EnumRange (OFF ON)
module.13.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.13.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.13.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.13.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.13.name	TEXT		StringRange
module.13.present	TEXT	R/O	StringRange R/O
module.13.type	TEXT	R/O	StringRange R/O
module.13.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.13.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.13.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.13.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
module.13.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.13.voltageState	TEXT	R/O	StringRange R/O
module.13.currentState	TEXT	R/O	StringRange R/O
module.13.rfLevelState	TEXT	R/O	StringRange R/O
module.13.ampVoltageState	TEXT	R/O	StringRange R/O
module.13.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.13.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.13.tempState	TEXT	R/O	StringRange R/O
module.14.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.14.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.14.22kHz	CHOICE		EnumRange (OFF ON)
module.14.reference	CHOICE		EnumRange (OFF ON)
module.14.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.14.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.14.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.14.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.14.name	TEXT		StringRange
module.14.present	TEXT	R/O	StringRange R/O
module.14.type	TEXT	R/O	StringRange R/O
module.14.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.14.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.14.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.14.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.14.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.14.voltageState	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.14.currentState	TEXT	R/O	StringRange R/O
module.14.rfLevelState	TEXT	R/O	StringRange R/O
module.14.ampVoltageState	TEXT	R/O	StringRange R/O
module.14.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.14.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.14.tempState	TEXT	R/O	StringRange R/O
module.15.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.15.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.15.22kHz	CHOICE		EnumRange (OFF ON)
module.15.reference	CHOICE		EnumRange (OFF ON)
module.15.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.15.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.15.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.15.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.15.name	TEXT		StringRange
module.15.present	TEXT	R/O	StringRange R/O
module.15.type	TEXT	R/O	StringRange R/O
module.15.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.15.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.15.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.15.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.15.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.15.voltageState	TEXT	R/O	StringRange R/O
module.15.currentState	TEXT	R/O	StringRange R/O
module.15.rfLevelState	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
module.15.ampVoltageState	TEXT	R/O	StringRange R/O
module.15.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)
module.15.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.15.tempState	TEXT	R/O	StringRange R/O
module.16.voltage	CHOICE		EnumRange (OFF CUSTOM TRACKING 3.3V 5V 8V 13V 15V 18V 20V 24V 36V 48V)
module.16.customVoltage	FLOAT		DoubleRange (0.0 .. 21.0)
module.16.22kHz	CHOICE		EnumRange (OFF ON)
module.16.reference	CHOICE		EnumRange (OFF ON)
module.16.currentLowLimit	INTEGER		IntegerRange (0 .. 6000)
module.16.currentHighLimit	INTEGER		IntegerRange (0 .. 6000)
module.16.rfLevelLowLimit	INTEGER		IntegerRange (-100 .. 0)
module.16.rfLevelHighLimit	INTEGER		IntegerRange (-100 .. 0)
module.16.name	TEXT		StringRange
module.16.present	TEXT	R/O	StringRange R/O
module.16.type	TEXT	R/O	StringRange R/O
module.16.meas.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.16.meas.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.16.meas.power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.16.meas.rfLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.16.meas.ampVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
module.16.voltageState	TEXT	R/O	StringRange R/O
module.16.currentState	TEXT	R/O	StringRange R/O
module.16.rfLevelState	TEXT	R/O	StringRange R/O
module.16.ampVoltageState	TEXT	R/O	StringRange R/O
module.16.sumState	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
module.16.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
module.16.tempState	TEXT	R/O	StringRange R/O
internal.01.present	TEXT	R/O	StringRange R/O
internal.02.present	TEXT	R/O	StringRange R/O
internal.03.present	TEXT	R/O	StringRange R/O
internal.04.present	TEXT	R/O	StringRange R/O
internal.05.present	TEXT	R/O	StringRange R/O
internal.06.present	TEXT	R/O	StringRange R/O
internal.07.present	TEXT	R/O	StringRange R/O
internal.08.present	TEXT	R/O	StringRange R/O
internal.09.present	TEXT	R/O	StringRange R/O
internal.10.present	TEXT	R/O	StringRange R/O
internal.11.present	TEXT	R/O	StringRange R/O
internal.12.present	TEXT	R/O	StringRange R/O
internal.13.present	TEXT	R/O	StringRange R/O
internal.14.present	TEXT	R/O	StringRange R/O
internal.15.present	TEXT	R/O	StringRange R/O
internal.16.present	TEXT	R/O	StringRange R/O
internal.psu.1.present	TEXT	R/O	StringRange R/O
internal.psu.2.present	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Module 1 Voltage)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Module 1 Current)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 1 RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 1 Amplifier)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Module 1 Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Module 2 Voltage)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Module 2 Current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Module 2 RF)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module 2 Amplifier)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module 2 Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Module 3 Voltage)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Module 3 Current)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Module 3 RF)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Module 3 Amplifier)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Module 3 Temperature)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Module 4 Voltage)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Module 4 Current)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Module 4 RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Module 4 Amplifier)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Module 4 Temperature)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (Module 5 Voltage)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Module 5 Current)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Module 5 RF)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Module 5 Amplifier)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Module 5 Temperature)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Module 6 Voltage)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Module 6 Current)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Module 6 RF)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Module 6 Amplifier)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Module 6 Temperature)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Module 7 Voltage)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Module 7 Current)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Module 7 RF)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Module 7 Amplifier)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Module 7 Temperature)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Module 8 Voltage)

name	*type*	*flags*	*range*
faults.37	ALARM	R/O	AlarmFlagRange R/O (Module 8 Current)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Module 8 RF)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Module 8 Amplifier)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Module 8 Temperature)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Module 9 Voltage)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Module 9 Current)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Module 9 RF)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Module 9 Amplifier)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Module 9 Temperature)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Module 10 Voltage)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Module 10 Current)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Module 10 RF)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Module 10 Amplifier)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Module 10 Temperature)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Module 11 Voltage)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Module 11 Current)

name	*type*	*flags*	*range*
faults.53	ALARM	R/O	AlarmFlagRange R/O (Module 11 RF)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Module 11 Amplifier)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Module 11 Temperature)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Module 12 Voltage)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Module 12 Current)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Module 12 RF)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Module 12 Amplifier)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Module 12 Temperature)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Module 13 Voltage)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Module 13 Current)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Module 13 RF)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Module 13 Amplifier)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Module 13 Temperature)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Module 14 Voltage)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Module 14 Current)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Module 14 RF)

name	*type*	*flags*	*range*
faults.69	ALARM	R/O	AlarmFlagRange R/O (Module 14 Amplifier)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Module 14 Temperature)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Module 15 Voltage)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Module 15 Current)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Module 15 RF)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Module 15 Amplifier)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Module 15 Temperature)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Module 16 Voltage)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Module 16 Current)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Module 16 RF)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Module 16 Amplifier)
faults.80	ALARM	R/O	AlarmFlagRange R/O (Module 16 Temperature)
faults.81	ALARM	R/O	AlarmFlagRange R/O (10 MHz)
faults.82	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.83	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.84	ALARM	R/O	AlarmFlagRange R/O (PSU 1 temperature)
faults.85	ALARM	R/O	AlarmFlagRange R/O (PSU 2 temperature)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Chassis Summary)

9.213 ETL-SHF-Switch-2to2

This page describes the device driver and the device window for etl systems 2x2 SHF Switch.

It supports models from the followings series: 23264

SW Version:

- e413 Version 2.4

To allow proper operation the exact model and number of inputs has to be defined in the configuration page

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device including the number of inputs and device model.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
switch	Defines the switch in the SHF chassis to be controlled
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.noSwitches	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.switch	CHOICE	R/O	EnumRange R/O (1 2)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (A B)
location	CHOICE	R/O	EnumRange R/O (1 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU Bus)

9.214 ETL-SHF-Switch

This page describes the device driver and the device window for etl systems 8x1, 16x1 or 32x1 SHF Switch.

It supports models from the followings series: 23257, 23259 and 23260

To allow proper operation the exact model and number of inputs has to be defined in the configuration page

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device including the number of inputs and device model.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port

number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- model --- Defines model series. This is necessary for proper communication.

- inputs --- Defines number of inputs. This is necessary for proper communication.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.model	CHOICE	SAVE SETUP	EnumRange (23257 23259 23260)
config.inputs	CHOICE	SAVE SETUP	EnumRange (8 16 32)
info.model	TEXT	R/O	StringRange R/O
position	CHOICE		EnumRange (0)
internal.oid	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU Bus)

9.215 ETL-Splitter-DIV409

This page describes the device driver and the device window for etl systems splitter card DIV 409.

Together with a etl system 26128 frame each DIV 409 card has to be addressed individually.

To allow proper operation the slot number and device address has to be defined in the configuration page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	Must always be set to B if the device is controlled TCP or over a serial line.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
slot	Defines the slot number of the respective module. This is necessary for proper communication.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
voltage.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)
protection	CHOICE		EnumRange (ON OFF)
number.amp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.01.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.02.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.03.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.04.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
number.attn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
attn.01	FLOAT		DoubleRange (0.0 .. 8.0)

name	*type*	*flags*	*range*
attn.02	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Card status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Attenuator 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Attenuator 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.216 ETL-Splitter-DIV429

This page describes the device driver and the device window for etl systems splitter card DIV 429.

Together with a etl system 26128 frame each DIV 429 card has to be addressed individually.

To allow proper operation the slot number and device address has to be defined in the configuration page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	Must always be set to B if the device is controlled TCP or over a serial line.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
slot	Defines the slot number of the respective module. This is necessary for proper communication.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage8V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
config.rfLowThd	INTEGER	SETUP	IntegerRange (0 .. 0)
config.rfHighThd	INTEGER	SETUP	IntegerRange (0 .. 0)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
protection	CHOICE		EnumRange (ON OFF)
number.amp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
amp.01.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.02.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.03.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.04.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.05.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.06.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.07.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.08.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.09.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
number.attn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
attn.01	FLOAT		DoubleRange (0.0 .. 8.0)
attn.01.bias	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slope	FLOAT		DoubleRange (0.0 .. 0.0)
rf.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Card status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amplifier 5)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amplifier 6)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amplifier 7)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amplifier 8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amplifier 9)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Attenuator 1)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF Level)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.217 ETL-Stingray-FO-Base

This page describes the *ETL-Stingray-Base* device driver and its device window. The*ETL-Stingray-Base* device driver monitors and controls the base unit features and alarms of one ETL Stingray FO chassis. Beside the possibility to reset the unit this driver works as read-only.

Beside this base unit multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The drivers was tested with the following equipment:

- Model: SRY-C201-2U (16-card/2RU model)
- SW version: e363 2V3

The different available fibre optical cards of the ETL stingray series are:

- ETL-Stingray-FO-TX monitors and controls one single or dual channel FO transmit card.
- ETL-Stingray-FO-RX monitors and controls one single or dual channel FO receive card. Th Base unit as well as the available cards are prepared to support up to 16 individual FO cards.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification

consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mainCpuVersion	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fan1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.vbusVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.6.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fanPsu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fanPsu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.slot.01.name	TEXT	R/O	StringRange R/O
info.slot.02.name	TEXT	R/O	StringRange R/O
info.slot.03.name	TEXT	R/O	StringRange R/O
info.slot.04.name	TEXT	R/O	StringRange R/O
info.slot.05.name	TEXT	R/O	StringRange R/O
info.slot.06.name	TEXT	R/O	StringRange R/O
info.slot.07.name	TEXT	R/O	StringRange R/O
info.slot.08.name	TEXT	R/O	StringRange R/O
info.slot.09.name	TEXT	R/O	StringRange R/O
info.slot.10.name	TEXT	R/O	StringRange R/O
info.slot.11.name	TEXT	R/O	StringRange R/O
info.slot.12.name	TEXT	R/O	StringRange R/O
info.slot.13.name	TEXT	R/O	StringRange R/O
info.slot.14.name	TEXT	R/O	StringRange R/O
info.slot.15.name	TEXT	R/O	StringRange R/O
info.slot.16.name	TEXT	R/O	StringRange R/O
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
config.setLimits	CHOICE		EnumRange (ON OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Internal Comms)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (VBUS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (6.5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature)

9.218 ETL-Stingray-FO-RX

This page describes the *ETL-Stingray-FO-RX* device driver and its device window. The *ETL-Stingray-FO-RX* device driver monitors and controls one fibre receive card within a Stingray FO chassis.

Multiple instances of these device may be configured in the sat-nms M&C to handle all cards within the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical cards in the Stingray chassis this particular sat-nms device shall control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The rf and optical parameters.
- --- The amplifier voltage measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- setLimits --- En- or disable the monitoring functions of the card.
- fixGain.1 --- En- or disable fixed gain for channel 1.
- fixGain.2 --- En- or disable fixed gain for channel 2.
- rfPowerSet.1 --- Sets the rf output power (AGC) of channel 1.
- rfPowerSet.2 --- Sets the rf output power (AGC) of channel 2.
- hasDual --- Selects if the rx card is dual or single channel and en- or disable parameters accordingly.
- slot --- Selects the slot position where the controlled card is equipped. This is mandatory for proper operation.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.6.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.busVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.busCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.hasDual	CHOICE	SAVE SETUP	EnumRange (SINGLE DUAL)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
fixGain.1	CHOICE		EnumRange (OFF ON)
fixGain.2	CHOICE		EnumRange (OFF ON)
rfPowerSet.1	FLOAT		DoubleRange (0.00 .. 0.00)
rfPowerSet.2	FLOAT		DoubleRange (0.00 .. 0.00)
amp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.6	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RF 1 Output)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RF 2 Output)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Opt. 1 Input)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Opt. 2 Input)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (6.5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 5)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amplifier 6)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Bus Voltage)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Bus Current)

9.219 ETL-Stingray-FO-TX

This page describes the *ETL-Stingray-FO-TX* device driver and its device window. The *ETL-Stingray-FO-TX* device driver monitors and controls one fibre transmit card within a Stingray FO chassis.

Multiple instances of these device may be configured in the sat-nms M&C to handle all cards within the chassis. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'slot' configuration parameter is used to select which of the physical cards in the Stingray chassis this particular sat-nms device shall control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The rf and optical parameters.
- --- The amplifier voltage measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- setLimits --- En- or disable the monitoring functions of the card.
- fixGain.1 --- En- or disable fixed gain for channel 1.
- fixGain.2 --- En- or disable fixed gain for channel 2.
- rfPowerSet.1 --- Sets the rf output power (AGC) of channel 1.
- rfPowerSet.2 --- Sets the rf output power (AGC) of channel 2.
- lnbVoltage.1 --- Sets the lnb voltage of the rf input of channel 1.
- lnbVoltage.2 --- Sets the lnb voltage of the rf input of channel 2.
- 22kHz.1 --- Sets the 22kHz / tone of the rf input of channel 1.
- 22kHz.2 --- Sets the 22kHz / tone of the rf input of channel 2.
- limitsLnbCurr.1 --- En- or disable LNB current monitoring for channel 1. LNB current out of its +/- 20% range will set an alarm.
- limitsLnbCurr.2 --- En- or disable LNB current monitoring for channel 2. LNB current out of its +/- 20% range will set an alarm.
- hasLnb --- Selects if the tx card is equipped with LNB supply.
- hasDual --- Selects if the tx card is dual or single channel and en- or disable parameters accordingly.
- slot --- Selects the slot position where the controlled card is equipped. This is mandatory for proper operation.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.6.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.neg5Volt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.busVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.busCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
config.setLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.hasLnb	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.hasDual	CHOICE	SAVE SETUP	EnumRange (SINGLE DUAL)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
amp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
amp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.6	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.7	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.8	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.9	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
amp.10	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fixGain.1	CHOICE		EnumRange (OFF ON)
fixGain.2	CHOICE		EnumRange (OFF ON)
rfPowerSet.1	FLOAT		DoubleRange (0.00 .. 0.00)
rfPowerSet.2	FLOAT		DoubleRange (0.00 .. 0.00)
meas.lnbVoltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.lnbVoltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.lnbCurrent.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.lnbCurrent.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnbVoltage.1	CHOICE		EnumRange (OFF 13V 18V)
lnbVoltage.2	CHOICE		EnumRange (OFF 13V 18V)
lnb22kHz.1	CHOICE		EnumRange (OFF ON)
lnb22kHz.2	CHOICE		EnumRange (OFF ON)
limitsLnbCurr.1	CHOICE		EnumRange (OFF ON)
limitsLnbCurr.2	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RF 1 Input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RF 2 Input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Opt. 1 Output)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Opt. 2 Output)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (6.5V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.08	ALARM	R/O	AlarmFlagRange R/O (-5V)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amplifier 5)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Amplifier 6)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Amplifier 7)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Amplifier 8)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Amplifier 9)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Amplifier 10)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Bus Voltage)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Bus Current)
faults.23	ALARM	R/O	AlarmFlagRange R/O (LNB 1 Voltage)
faults.24	ALARM	R/O	AlarmFlagRange R/O (LNB 2 Voltage)
faults.25	ALARM	R/O	AlarmFlagRange R/O (LNB 1 Current)
faults.26	ALARM	R/O	AlarmFlagRange R/O (LNB 2 Current)

9.220 ETL-SwitchUnit-23258

This page describes the device driver and the device window for etl systems 16 channel 2:2 switch unit type 23258.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.unitTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psuBusVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fan1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.sw01.comment	TEXT	SAVE SETUP	StringRange
config.sw02.comment	TEXT	SAVE SETUP	StringRange
config.sw03.comment	TEXT	SAVE SETUP	StringRange
config.sw04.comment	TEXT	SAVE SETUP	StringRange
config.sw05.comment	TEXT	SAVE SETUP	StringRange
config.sw06.comment	TEXT	SAVE SETUP	StringRange
config.sw07.comment	TEXT	SAVE SETUP	StringRange
config.sw08.comment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.sw09.comment	TEXT	SAVE SETUP	StringRange
config.sw10.comment	TEXT	SAVE SETUP	StringRange
config.sw11.comment	TEXT	SAVE SETUP	StringRange
config.sw12.comment	TEXT	SAVE SETUP	StringRange
config.sw13.comment	TEXT	SAVE SETUP	StringRange
config.sw14.comment	TEXT	SAVE SETUP	StringRange
config.sw15.comment	TEXT	SAVE SETUP	StringRange
config.sw16.comment	TEXT	SAVE SETUP	StringRange
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.write	INTEGER		IntegerRange (0 .. 0)
sw.01.position	CHOICE		EnumRange (A B)
sw.02.position	CHOICE		EnumRange (A B)
sw.03.position	CHOICE		EnumRange (A B)
sw.04.position	CHOICE		EnumRange (A B)
sw.05.position	CHOICE		EnumRange (A B)
sw.06.position	CHOICE		EnumRange (A B)
sw.07.position	CHOICE		EnumRange (A B)
sw.08.position	CHOICE		EnumRange (A B)
sw.09.position	CHOICE		EnumRange (A B)
sw.10.position	CHOICE		EnumRange (A B)
sw.11.position	CHOICE		EnumRange (A B)
sw.12.position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
sw.13.position	CHOICE		EnumRange (A B)
sw.14.position	CHOICE		EnumRange (A B)
sw.15.position	CHOICE		EnumRange (A B)
sw.16.position	CHOICE		EnumRange (A B)
sw.01.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.02.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.03.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.04.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.05.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.06.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.07.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.08.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.09.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.10.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.11.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.12.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.13.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.14.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.15.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
sw.16.counter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU Bus)

9.221 ETL-VTR70-Matrix

This page describes the device driver and the device window for ETL L-Band Matrix Victor series VTR-70. The driver works with the distributive model.

It supports 4x4 up to 16x16 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching by name 1-16.
- --- Port switching by number 1-16.
- --- Input settings and readings.
- --- Edit in and output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- rfCardMon --- Allows to en- or disable the rf card monitoring of the matrix.
- chassisMon --- Allows to en- or disable the chassis monitoring of the matrix.

- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rf1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rf2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fan1Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan2Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan3Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan4Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.rfFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.rfCardMon	CHOICE	SETUP	EnumRange (ON OFF)
config.chassisMon	CHOICE	SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
aliasRead	TEXT	NOPRESET	StringRange
reboot	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.01.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.02.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.03.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.04.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
input.05.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.06.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.07.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.08.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.09.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.10.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.11.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.12.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.13.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.14.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.15.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.16.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
input.01.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.02.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.03.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.04.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.05.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.06.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.07.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.08.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)

name	*type*	*flags*	*range*
input.09.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.10.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.11.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.12.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.13.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.14.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.15.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.16.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
input.01.lnbTone	CHOICE		EnumRange (OFF ON)
input.02.lnbTone	CHOICE		EnumRange (OFF ON)
input.03.lnbTone	CHOICE		EnumRange (OFF ON)
input.04.lnbTone	CHOICE		EnumRange (OFF ON)
input.05.lnbTone	CHOICE		EnumRange (OFF ON)
input.06.lnbTone	CHOICE		EnumRange (OFF ON)
input.07.lnbTone	CHOICE		EnumRange (OFF ON)
input.08.lnbTone	CHOICE		EnumRange (OFF ON)
input.09.lnbTone	CHOICE		EnumRange (OFF ON)
input.10.lnbTone	CHOICE		EnumRange (OFF ON)
input.11.lnbTone	CHOICE		EnumRange (OFF ON)
input.12.lnbTone	CHOICE		EnumRange (OFF ON)
input.13.lnbTone	CHOICE		EnumRange (OFF ON)
input.14.lnbTone	CHOICE		EnumRange (OFF ON)
input.15.lnbTone	CHOICE		EnumRange (OFF ON)
input.16.lnbTone	CHOICE		EnumRange (OFF ON)
input.01.lowLimit	INTEGER		IntegerRange (-50 .. 5)

name	*type*	*flags*	*range*
input.02.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.03.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.04.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.05.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.06.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.07.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.08.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.09.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.10.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.11.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.12.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.13.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.14.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.15.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.16.lowLimit	INTEGER		IntegerRange (-50 .. 5)
input.01.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.02.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.03.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.04.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.05.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.06.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.07.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.08.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.09.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.10.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.11.highLimit	INTEGER		IntegerRange (-50 .. 5)

name	*type*	*flags*	*range*
input.12.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.13.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.14.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.15.highLimit	INTEGER		IntegerRange (-50 .. 5)
input.16.highLimit	INTEGER		IntegerRange (-50 .. 5)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
output.01.locking	INTEGER		IntegerRange (0 .. 1)
output.02.locking	INTEGER		IntegerRange (0 .. 1)
output.03.locking	INTEGER		IntegerRange (0 .. 1)
output.04.locking	INTEGER		IntegerRange (0 .. 1)
output.05.locking	INTEGER		IntegerRange (0 .. 1)
output.06.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.07.locking	INTEGER		IntegerRange (0 .. 1)
output.08.locking	INTEGER		IntegerRange (0 .. 1)
output.09.locking	INTEGER		IntegerRange (0 .. 1)
output.10.locking	INTEGER		IntegerRange (0 .. 1)
output.11.locking	INTEGER		IntegerRange (0 .. 1)
output.12.locking	INTEGER		IntegerRange (0 .. 1)
output.13.locking	INTEGER		IntegerRange (0 .. 1)
output.14.locking	INTEGER		IntegerRange (0 .. 1)
output.15.locking	INTEGER		IntegerRange (0 .. 1)
output.16.locking	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.10	ALARM	R/O	AlarmFlagRange R/O (CPU Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Chassis Communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF Card Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF Card 5V)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (RF Card -5V)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RF Card 1 Temperature)
faults.16	ALARM	R/O	AlarmFlagRange R/O (RF Card 2 Temperature)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Input 9)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Input 15)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Input 16)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Output 6)

name	*type*	*flags*	*range*
faults.47	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Output 15)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Output 16)

9.222 ETL-VTR70C-Matrix

This page describes the device driver and the device window for ETL L-Band Matrix Victor series VTR-70C.

The driver works with the "C" model (the combining matrix) only. It supports 4x4 up to 16x16 models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-16.
- 1 2 3 --- Port switching by number 1-16.
- --- Output settings and readings.
- --- Edit in and output names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- rfCardMon --- Allows to en- or disable the rf card monitoring of the matrix.
- chassisMon --- Allows to en- or disable the chassis monitoring of the matrix.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rf1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.rf2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fan1Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan2Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan3Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan4Speed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.rfFlt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp1Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.amp2Flt	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.rfCardMon	CHOICE	SETUP	EnumRange (ON OFF)

name	*type*	*flags*	*range*
config.chassisMon	CHOICE	SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
aliasRead	TEXT	NOPRESET	StringRange
reboot	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
output.01.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.02.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.03.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.04.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.05.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.06.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.07.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.08.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.09.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.10.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.11.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.12.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.13.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.14.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.15.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.16.rfLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.01	CHOICE		EnumRange (OFF)
input.02	CHOICE		EnumRange (OFF)
input.03	CHOICE		EnumRange (OFF)
input.04	CHOICE		EnumRange (OFF)
input.05	CHOICE		EnumRange (OFF)
input.06	CHOICE		EnumRange (OFF)
input.07	CHOICE		EnumRange (OFF)
input.08	CHOICE		EnumRange (OFF)
input.09	CHOICE		EnumRange (OFF)
input.10	CHOICE		EnumRange (OFF)
input.11	CHOICE		EnumRange (OFF)
input.12	CHOICE		EnumRange (OFF)
input.13	CHOICE		EnumRange (OFF)
input.14	CHOICE		EnumRange (OFF)
input.15	CHOICE		EnumRange (OFF)
input.16	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
output.01.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.02.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.03.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.04.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.05.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.06.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.07.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.08.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.09.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.10.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.11.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.12.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.13.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.14.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.15.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.16.lnbVoltage	CHOICE		EnumRange (OFF 13V 18V)
output.01.lnbTone	CHOICE		EnumRange (OFF ON)
output.02.lnbTone	CHOICE		EnumRange (OFF ON)
output.03.lnbTone	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
output.04.lnbTone	CHOICE		EnumRange (OFF ON)
output.05.lnbTone	CHOICE		EnumRange (OFF ON)
output.06.lnbTone	CHOICE		EnumRange (OFF ON)
output.07.lnbTone	CHOICE		EnumRange (OFF ON)
output.08.lnbTone	CHOICE		EnumRange (OFF ON)
output.09.lnbTone	CHOICE		EnumRange (OFF ON)
output.10.lnbTone	CHOICE		EnumRange (OFF ON)
output.11.lnbTone	CHOICE		EnumRange (OFF ON)
output.12.lnbTone	CHOICE		EnumRange (OFF ON)
output.13.lnbTone	CHOICE		EnumRange (OFF ON)
output.14.lnbTone	CHOICE		EnumRange (OFF ON)
output.15.lnbTone	CHOICE		EnumRange (OFF ON)
output.16.lnbTone	CHOICE		EnumRange (OFF ON)
output.01.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.02.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.03.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.04.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.05.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.06.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.07.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.08.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.09.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.10.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.11.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.12.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.13.lowLimit	INTEGER		IntegerRange (-50 .. 5)

name	*type*	*flags*	*range*
output.14.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.15.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.16.lowLimit	INTEGER		IntegerRange (-50 .. 5)
output.01.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.02.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.03.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.04.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.05.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.06.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.07.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.08.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.09.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.10.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.11.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.12.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.13.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.14.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.15.highLimit	INTEGER		IntegerRange (-50 .. 5)
output.16.highLimit	INTEGER		IntegerRange (-50 .. 5)
gain.01	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.02	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.03	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.04	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.05	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.06	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.07	FLOAT		DoubleRange (-5.0 .. 5.0)

name	*type*	*flags*	*range*
gain.08	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.09	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.10	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.11	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.12	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.13	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.14	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.15	FLOAT		DoubleRange (-5.0 .. 5.0)
gain.16	FLOAT		DoubleRange (-5.0 .. 5.0)
input.01.locking	INTEGER		IntegerRange (0 .. 1)
input.02.locking	INTEGER		IntegerRange (0 .. 1)
input.03.locking	INTEGER		IntegerRange (0 .. 1)
input.04.locking	INTEGER		IntegerRange (0 .. 1)
input.05.locking	INTEGER		IntegerRange (0 .. 1)
input.06.locking	INTEGER		IntegerRange (0 .. 1)
input.07.locking	INTEGER		IntegerRange (0 .. 1)
input.08.locking	INTEGER		IntegerRange (0 .. 1)
input.09.locking	INTEGER		IntegerRange (0 .. 1)
input.10.locking	INTEGER		IntegerRange (0 .. 1)
input.11.locking	INTEGER		IntegerRange (0 .. 1)
input.12.locking	INTEGER		IntegerRange (0 .. 1)
input.13.locking	INTEGER		IntegerRange (0 .. 1)
input.14.locking	INTEGER		IntegerRange (0 .. 1)
input.15.locking	INTEGER		IntegerRange (0 .. 1)
input.16.locking	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.10	ALARM	R/O	AlarmFlagRange R/O (CPU Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Chassis Communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF Card Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF Card 5V)
faults.14	ALARM	R/O	AlarmFlagRange R/O (RF Card -5V)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RF Card 1 Temperature)
faults.16	ALARM	R/O	AlarmFlagRange R/O (RF Card 2 Temperature)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Output 8)

name	*type*	*flags*	*range*
faults.29	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Output 15)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Output 16)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Input 9)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Input 15)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Input 16)

9.223 ETL-VTX100-Matrix

This page describes the *ETL-VTX-100-Matrix* device driver and its device window. The*ETL-VTX-*

100-Matrix device driver monitors and controls one distributive Vortex VTX-100 matrix with up to 64 inputs and 64 outputs.

As the number of installed in and outputs can not be read from the device, it is necessary to configure the matrix dimensions within the software using the device configuration page.

The driver was tested with the following equipment:

- VTX-101 16x16
  - SW version: e617 1v12

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by alias name 1-32
- abc --- Port switching by alias name 33-64
- 123 --- Port switching by number 1-32
- 123 --- Port switching by number 33-64
- --- Edit input alias names.
- --- Edit output alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.swReleaseDate	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.cpuTemperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu1Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu2Temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.psu1FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.psu2FanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fanCard1Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fanCard2Current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
aliasRead	TEXT	NOPRESET	StringRange
reboot	TEXT	NOPRESET	StringRange
resetFaults	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
output.33.name	TEXT		StringRange
output.34.name	TEXT		StringRange
output.35.name	TEXT		StringRange
output.36.name	TEXT		StringRange
output.37.name	TEXT		StringRange
output.38.name	TEXT		StringRange
output.39.name	TEXT		StringRange
output.40.name	TEXT		StringRange
output.41.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.42.name	TEXT		StringRange
output.43.name	TEXT		StringRange
output.44.name	TEXT		StringRange
output.45.name	TEXT		StringRange
output.46.name	TEXT		StringRange
output.47.name	TEXT		StringRange
output.48.name	TEXT		StringRange
output.49.name	TEXT		StringRange
output.50.name	TEXT		StringRange
output.51.name	TEXT		StringRange
output.52.name	TEXT		StringRange
output.53.name	TEXT		StringRange
output.54.name	TEXT		StringRange
output.55.name	TEXT		StringRange
output.56.name	TEXT		StringRange
output.57.name	TEXT		StringRange
output.58.name	TEXT		StringRange
output.59.name	TEXT		StringRange
output.60.name	TEXT		StringRange
output.61.name	TEXT		StringRange
output.62.name	TEXT		StringRange
output.63.name	TEXT		StringRange
output.64.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.30.name	TEXT		StringRange
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange
input.43.name	TEXT		StringRange
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange
input.55.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.56.name	TEXT		StringRange
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
output.17	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.18	CHOICE		EnumRange (OFF)
output.19	CHOICE		EnumRange (OFF)
output.20	CHOICE		EnumRange (OFF)
output.21	CHOICE		EnumRange (OFF)
output.22	CHOICE		EnumRange (OFF)
output.23	CHOICE		EnumRange (OFF)
output.24	CHOICE		EnumRange (OFF)
output.25	CHOICE		EnumRange (OFF)
output.26	CHOICE		EnumRange (OFF)
output.27	CHOICE		EnumRange (OFF)
output.28	CHOICE		EnumRange (OFF)
output.29	CHOICE		EnumRange (OFF)
output.30	CHOICE		EnumRange (OFF)
output.31	CHOICE		EnumRange (OFF)
output.32	CHOICE		EnumRange (OFF)
output.33	CHOICE		EnumRange (OFF)
output.34	CHOICE		EnumRange (OFF)
output.35	CHOICE		EnumRange (OFF)
output.36	CHOICE		EnumRange (OFF)
output.37	CHOICE		EnumRange (OFF)
output.38	CHOICE		EnumRange (OFF)
output.39	CHOICE		EnumRange (OFF)
output.40	CHOICE		EnumRange (OFF)
output.41	CHOICE		EnumRange (OFF)
output.42	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.43	CHOICE		EnumRange (OFF)
output.44	CHOICE		EnumRange (OFF)
output.45	CHOICE		EnumRange (OFF)
output.46	CHOICE		EnumRange (OFF)
output.47	CHOICE		EnumRange (OFF)
output.48	CHOICE		EnumRange (OFF)
output.49	CHOICE		EnumRange (OFF)
output.50	CHOICE		EnumRange (OFF)
output.51	CHOICE		EnumRange (OFF)
output.52	CHOICE		EnumRange (OFF)
output.53	CHOICE		EnumRange (OFF)
output.54	CHOICE		EnumRange (OFF)
output.55	CHOICE		EnumRange (OFF)
output.56	CHOICE		EnumRange (OFF)
output.57	CHOICE		EnumRange (OFF)
output.58	CHOICE		EnumRange (OFF)
output.59	CHOICE		EnumRange (OFF)
output.60	CHOICE		EnumRange (OFF)
output.61	CHOICE		EnumRange (OFF)
output.62	CHOICE		EnumRange (OFF)
output.63	CHOICE		EnumRange (OFF)
output.64	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
output.01.locking	INTEGER		IntegerRange (0 .. 1)
output.02.locking	INTEGER		IntegerRange (0 .. 1)
output.03.locking	INTEGER		IntegerRange (0 .. 1)
output.04.locking	INTEGER		IntegerRange (0 .. 1)
output.05.locking	INTEGER		IntegerRange (0 .. 1)
output.06.locking	INTEGER		IntegerRange (0 .. 1)
output.07.locking	INTEGER		IntegerRange (0 .. 1)
output.08.locking	INTEGER		IntegerRange (0 .. 1)
output.09.locking	INTEGER		IntegerRange (0 .. 1)
output.10.locking	INTEGER		IntegerRange (0 .. 1)
output.11.locking	INTEGER		IntegerRange (0 .. 1)
output.12.locking	INTEGER		IntegerRange (0 .. 1)
output.13.locking	INTEGER		IntegerRange (0 .. 1)
output.14.locking	INTEGER		IntegerRange (0 .. 1)
output.15.locking	INTEGER		IntegerRange (0 .. 1)
output.16.locking	INTEGER		IntegerRange (0 .. 1)
output.17.locking	INTEGER		IntegerRange (0 .. 1)
output.18.locking	INTEGER		IntegerRange (0 .. 1)
output.19.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.20.locking	INTEGER		IntegerRange (0 .. 1)
output.21.locking	INTEGER		IntegerRange (0 .. 1)
output.22.locking	INTEGER		IntegerRange (0 .. 1)
output.23.locking	INTEGER		IntegerRange (0 .. 1)
output.24.locking	INTEGER		IntegerRange (0 .. 1)
output.25.locking	INTEGER		IntegerRange (0 .. 1)
output.26.locking	INTEGER		IntegerRange (0 .. 1)
output.27.locking	INTEGER		IntegerRange (0 .. 1)
output.28.locking	INTEGER		IntegerRange (0 .. 1)
output.29.locking	INTEGER		IntegerRange (0 .. 1)
output.30.locking	INTEGER		IntegerRange (0 .. 1)
output.31.locking	INTEGER		IntegerRange (0 .. 1)
output.32.locking	INTEGER		IntegerRange (0 .. 1)
output.33.locking	INTEGER		IntegerRange (0 .. 1)
output.34.locking	INTEGER		IntegerRange (0 .. 1)
output.35.locking	INTEGER		IntegerRange (0 .. 1)
output.36.locking	INTEGER		IntegerRange (0 .. 1)
output.37.locking	INTEGER		IntegerRange (0 .. 1)
output.38.locking	INTEGER		IntegerRange (0 .. 1)
output.39.locking	INTEGER		IntegerRange (0 .. 1)
output.40.locking	INTEGER		IntegerRange (0 .. 1)
output.41.locking	INTEGER		IntegerRange (0 .. 1)
output.42.locking	INTEGER		IntegerRange (0 .. 1)
output.43.locking	INTEGER		IntegerRange (0 .. 1)
output.44.locking	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
output.45.locking	INTEGER		IntegerRange (0 .. 1)
output.46.locking	INTEGER		IntegerRange (0 .. 1)
output.47.locking	INTEGER		IntegerRange (0 .. 1)
output.48.locking	INTEGER		IntegerRange (0 .. 1)
output.49.locking	INTEGER		IntegerRange (0 .. 1)
output.50.locking	INTEGER		IntegerRange (0 .. 1)
output.51.locking	INTEGER		IntegerRange (0 .. 1)
output.52.locking	INTEGER		IntegerRange (0 .. 1)
output.53.locking	INTEGER		IntegerRange (0 .. 1)
output.54.locking	INTEGER		IntegerRange (0 .. 1)
output.55.locking	INTEGER		IntegerRange (0 .. 1)
output.56.locking	INTEGER		IntegerRange (0 .. 1)
output.57.locking	INTEGER		IntegerRange (0 .. 1)
output.58.locking	INTEGER		IntegerRange (0 .. 1)
output.59.locking	INTEGER		IntegerRange (0 .. 1)
output.60.locking	INTEGER		IntegerRange (0 .. 1)
output.61.locking	INTEGER		IntegerRange (0 .. 1)
output.62.locking	INTEGER		IntegerRange (0 .. 1)
output.63.locking	INTEGER		IntegerRange (0 .. 1)
output.64.locking	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Chassis Communication)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU 1)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature CPU)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan PSU 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan Card 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan Card 2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input Card 1-4)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input Card 5-8)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input Card 9-12)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input Card 13-16)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input Card 17-20)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Input Card 21-24)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Input Card 25-28)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Input Card 29-32)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Input Card 33-36)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Input Card 37-40)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Input Card 41-44)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (Input Card 45-48)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Input Card 49-52)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Input Card 53-56)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Input Card 57-60)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Input Card 61-64)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Output Card 1-4)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Output Card 5-8)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Output Card 9-12)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Output Card 13-16)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Output Card 17-20)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Output Card 21-24)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Output Card 25-28)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Output Card 29-32)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Output Card 33-36)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Output Card 37-40)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Output Card 41-44)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Output Card 45-48)

name	*type*	*flags*	*range*
faults.53	ALARM	R/O	AlarmFlagRange R/O (Output Card 49-52)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Output Card 53-56)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Output Card 57-60)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Output Card 61-64)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 1)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 2)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 3)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 4)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 5)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 6)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 7)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 8)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 9)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 10)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 11)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 12)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 13)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 14)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 15)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Matrix Card 16)

9.224 ETL-Volcano-Matrix

This page describes the device driver and the device window for ETL L-Band Switch Matrix Volcano with series 128 inputs and 512 outputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-100.
- abc --- Port switching by name 101-200.
- abc --- Port switching by name 201-300.
- abc --- Port switching by name 301-400.
- abc --- Port switching by name 401-500.
- abc --- Port switching by name 501-512.
- 123 --- Port switching by number 1-100.
- 123 --- Port switching by number 101-200.
- 123 --- Port switching by number 201-300.
- 123 --- Port switching by number 301-400.
- 123 --- Port switching by number 401-500.
- 123 --- Port switching by number 501-512.
- --- Edit input names 1-128.
- --- Edit output names 1-100.
- --- Edit output names 101-100.
- --- Edit output names 201-300.
- --- Edit output names 301-400.
- --- Edit output names 401-500.
- --- Edit output names 5011-512.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as the hot/cold standby settings per module.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the READ ALIAS NAMES FROM CHASSIS button on the device driver. The reason is that reading these value is only done initially at the server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.cpuVersion	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
aliasRead	TEXT		StringRange
reset	TEXT	NOPRESET	StringRange
output.001.name	TEXT		StringRange
output.002.name	TEXT		StringRange
output.003.name	TEXT		StringRange
output.004.name	TEXT		StringRange
output.005.name	TEXT		StringRange
output.006.name	TEXT		StringRange
output.007.name	TEXT		StringRange
output.008.name	TEXT		StringRange
output.009.name	TEXT		StringRange
output.010.name	TEXT		StringRange
output.011.name	TEXT		StringRange
output.012.name	TEXT		StringRange
output.013.name	TEXT		StringRange
output.014.name	TEXT		StringRange
output.015.name	TEXT		StringRange
output.016.name	TEXT		StringRange
output.017.name	TEXT		StringRange
output.018.name	TEXT		StringRange
output.019.name	TEXT		StringRange
output.020.name	TEXT		StringRange
output.021.name	TEXT		StringRange
output.022.name	TEXT		StringRange
output.023.name	TEXT		StringRange
output.024.name	TEXT		StringRange
output.025.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.026.name	TEXT		StringRange
output.027.name	TEXT		StringRange
output.028.name	TEXT		StringRange
output.029.name	TEXT		StringRange
output.030.name	TEXT		StringRange
output.031.name	TEXT		StringRange
output.032.name	TEXT		StringRange
output.033.name	TEXT		StringRange
output.034.name	TEXT		StringRange
output.035.name	TEXT		StringRange
output.036.name	TEXT		StringRange
output.037.name	TEXT		StringRange
output.038.name	TEXT		StringRange
output.039.name	TEXT		StringRange
output.040.name	TEXT		StringRange
output.041.name	TEXT		StringRange
output.042.name	TEXT		StringRange
output.043.name	TEXT		StringRange
output.044.name	TEXT		StringRange
output.045.name	TEXT		StringRange
output.046.name	TEXT		StringRange
output.047.name	TEXT		StringRange
output.048.name	TEXT		StringRange
output.049.name	TEXT		StringRange
output.050.name	TEXT		StringRange
output.051.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.052.name	TEXT		StringRange
output.053.name	TEXT		StringRange
output.054.name	TEXT		StringRange
output.055.name	TEXT		StringRange
output.056.name	TEXT		StringRange
output.057.name	TEXT		StringRange
output.058.name	TEXT		StringRange
output.059.name	TEXT		StringRange
output.060.name	TEXT		StringRange
output.061.name	TEXT		StringRange
output.062.name	TEXT		StringRange
output.063.name	TEXT		StringRange
output.064.name	TEXT		StringRange
output.065.name	TEXT		StringRange
output.066.name	TEXT		StringRange
output.067.name	TEXT		StringRange
output.068.name	TEXT		StringRange
output.069.name	TEXT		StringRange
output.070.name	TEXT		StringRange
output.071.name	TEXT		StringRange
output.072.name	TEXT		StringRange
output.073.name	TEXT		StringRange
output.074.name	TEXT		StringRange
output.075.name	TEXT		StringRange
output.076.name	TEXT		StringRange
output.077.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.078.name	TEXT		StringRange
output.079.name	TEXT		StringRange
output.080.name	TEXT		StringRange
output.081.name	TEXT		StringRange
output.082.name	TEXT		StringRange
output.083.name	TEXT		StringRange
output.084.name	TEXT		StringRange
output.085.name	TEXT		StringRange
output.086.name	TEXT		StringRange
output.087.name	TEXT		StringRange
output.088.name	TEXT		StringRange
output.089.name	TEXT		StringRange
output.090.name	TEXT		StringRange
output.091.name	TEXT		StringRange
output.092.name	TEXT		StringRange
output.093.name	TEXT		StringRange
output.094.name	TEXT		StringRange
output.095.name	TEXT		StringRange
output.096.name	TEXT		StringRange
output.097.name	TEXT		StringRange
output.098.name	TEXT		StringRange
output.099.name	TEXT		StringRange
output.100.name	TEXT		StringRange
output.101.name	TEXT		StringRange
output.102.name	TEXT		StringRange
output.103.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.104.name	TEXT		StringRange
output.105.name	TEXT		StringRange
output.106.name	TEXT		StringRange
output.107.name	TEXT		StringRange
output.108.name	TEXT		StringRange
output.109.name	TEXT		StringRange
output.110.name	TEXT		StringRange
output.111.name	TEXT		StringRange
output.112.name	TEXT		StringRange
output.113.name	TEXT		StringRange
output.114.name	TEXT		StringRange
output.115.name	TEXT		StringRange
output.116.name	TEXT		StringRange
output.117.name	TEXT		StringRange
output.118.name	TEXT		StringRange
output.119.name	TEXT		StringRange
output.120.name	TEXT		StringRange
output.121.name	TEXT		StringRange
output.122.name	TEXT		StringRange
output.123.name	TEXT		StringRange
output.124.name	TEXT		StringRange
output.125.name	TEXT		StringRange
output.126.name	TEXT		StringRange
output.127.name	TEXT		StringRange
output.128.name	TEXT		StringRange
output.129.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.130.name	TEXT		StringRange
output.131.name	TEXT		StringRange
output.132.name	TEXT		StringRange
output.133.name	TEXT		StringRange
output.134.name	TEXT		StringRange
output.135.name	TEXT		StringRange
output.136.name	TEXT		StringRange
output.137.name	TEXT		StringRange
output.138.name	TEXT		StringRange
output.139.name	TEXT		StringRange
output.140.name	TEXT		StringRange
output.141.name	TEXT		StringRange
output.142.name	TEXT		StringRange
output.143.name	TEXT		StringRange
output.144.name	TEXT		StringRange
output.145.name	TEXT		StringRange
output.146.name	TEXT		StringRange
output.147.name	TEXT		StringRange
output.148.name	TEXT		StringRange
output.149.name	TEXT		StringRange
output.150.name	TEXT		StringRange
output.151.name	TEXT		StringRange
output.152.name	TEXT		StringRange
output.153.name	TEXT		StringRange
output.154.name	TEXT		StringRange
output.155.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.156.name	TEXT		StringRange
output.157.name	TEXT		StringRange
output.158.name	TEXT		StringRange
output.159.name	TEXT		StringRange
output.160.name	TEXT		StringRange
output.161.name	TEXT		StringRange
output.162.name	TEXT		StringRange
output.163.name	TEXT		StringRange
output.164.name	TEXT		StringRange
output.165.name	TEXT		StringRange
output.166.name	TEXT		StringRange
output.167.name	TEXT		StringRange
output.168.name	TEXT		StringRange
output.169.name	TEXT		StringRange
output.170.name	TEXT		StringRange
output.171.name	TEXT		StringRange
output.172.name	TEXT		StringRange
output.173.name	TEXT		StringRange
output.174.name	TEXT		StringRange
output.175.name	TEXT		StringRange
output.176.name	TEXT		StringRange
output.177.name	TEXT		StringRange
output.178.name	TEXT		StringRange
output.179.name	TEXT		StringRange
output.180.name	TEXT		StringRange
output.181.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.182.name	TEXT		StringRange
output.183.name	TEXT		StringRange
output.184.name	TEXT		StringRange
output.185.name	TEXT		StringRange
output.186.name	TEXT		StringRange
output.187.name	TEXT		StringRange
output.188.name	TEXT		StringRange
output.189.name	TEXT		StringRange
output.190.name	TEXT		StringRange
output.191.name	TEXT		StringRange
output.192.name	TEXT		StringRange
output.193.name	TEXT		StringRange
output.194.name	TEXT		StringRange
output.195.name	TEXT		StringRange
output.196.name	TEXT		StringRange
output.197.name	TEXT		StringRange
output.198.name	TEXT		StringRange
output.199.name	TEXT		StringRange
output.200.name	TEXT		StringRange
output.201.name	TEXT		StringRange
output.202.name	TEXT		StringRange
output.203.name	TEXT		StringRange
output.204.name	TEXT		StringRange
output.205.name	TEXT		StringRange
output.206.name	TEXT		StringRange
output.207.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.208.name	TEXT		StringRange
output.209.name	TEXT		StringRange
output.210.name	TEXT		StringRange
output.211.name	TEXT		StringRange
output.212.name	TEXT		StringRange
output.213.name	TEXT		StringRange
output.214.name	TEXT		StringRange
output.215.name	TEXT		StringRange
output.216.name	TEXT		StringRange
output.217.name	TEXT		StringRange
output.218.name	TEXT		StringRange
output.219.name	TEXT		StringRange
output.220.name	TEXT		StringRange
output.221.name	TEXT		StringRange
output.222.name	TEXT		StringRange
output.223.name	TEXT		StringRange
output.224.name	TEXT		StringRange
output.225.name	TEXT		StringRange
output.226.name	TEXT		StringRange
output.227.name	TEXT		StringRange
output.228.name	TEXT		StringRange
output.229.name	TEXT		StringRange
output.230.name	TEXT		StringRange
output.231.name	TEXT		StringRange
output.232.name	TEXT		StringRange
output.233.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.234.name	TEXT		StringRange
output.235.name	TEXT		StringRange
output.236.name	TEXT		StringRange
output.237.name	TEXT		StringRange
output.238.name	TEXT		StringRange
output.239.name	TEXT		StringRange
output.240.name	TEXT		StringRange
output.241.name	TEXT		StringRange
output.242.name	TEXT		StringRange
output.243.name	TEXT		StringRange
output.244.name	TEXT		StringRange
output.245.name	TEXT		StringRange
output.246.name	TEXT		StringRange
output.247.name	TEXT		StringRange
output.248.name	TEXT		StringRange
output.249.name	TEXT		StringRange
output.250.name	TEXT		StringRange
output.251.name	TEXT		StringRange
output.252.name	TEXT		StringRange
output.253.name	TEXT		StringRange
output.254.name	TEXT		StringRange
output.255.name	TEXT		StringRange
output.256.name	TEXT		StringRange
output.257.name	TEXT		StringRange
output.258.name	TEXT		StringRange
output.259.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.260.name	TEXT		StringRange
output.261.name	TEXT		StringRange
output.262.name	TEXT		StringRange
output.263.name	TEXT		StringRange
output.264.name	TEXT		StringRange
output.265.name	TEXT		StringRange
output.266.name	TEXT		StringRange
output.267.name	TEXT		StringRange
output.268.name	TEXT		StringRange
output.269.name	TEXT		StringRange
output.270.name	TEXT		StringRange
output.271.name	TEXT		StringRange
output.272.name	TEXT		StringRange
output.273.name	TEXT		StringRange
output.274.name	TEXT		StringRange
output.275.name	TEXT		StringRange
output.276.name	TEXT		StringRange
output.277.name	TEXT		StringRange
output.278.name	TEXT		StringRange
output.279.name	TEXT		StringRange
output.280.name	TEXT		StringRange
output.281.name	TEXT		StringRange
output.282.name	TEXT		StringRange
output.283.name	TEXT		StringRange
output.284.name	TEXT		StringRange
output.285.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.286.name	TEXT		StringRange
output.287.name	TEXT		StringRange
output.288.name	TEXT		StringRange
output.289.name	TEXT		StringRange
output.290.name	TEXT		StringRange
output.291.name	TEXT		StringRange
output.292.name	TEXT		StringRange
output.293.name	TEXT		StringRange
output.294.name	TEXT		StringRange
output.295.name	TEXT		StringRange
output.296.name	TEXT		StringRange
output.297.name	TEXT		StringRange
output.298.name	TEXT		StringRange
output.299.name	TEXT		StringRange
output.300.name	TEXT		StringRange
output.301.name	TEXT		StringRange
output.302.name	TEXT		StringRange
output.303.name	TEXT		StringRange
output.304.name	TEXT		StringRange
output.305.name	TEXT		StringRange
output.306.name	TEXT		StringRange
output.307.name	TEXT		StringRange
output.308.name	TEXT		StringRange
output.309.name	TEXT		StringRange
output.310.name	TEXT		StringRange
output.311.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.312.name	TEXT		StringRange
output.313.name	TEXT		StringRange
output.314.name	TEXT		StringRange
output.315.name	TEXT		StringRange
output.316.name	TEXT		StringRange
output.317.name	TEXT		StringRange
output.318.name	TEXT		StringRange
output.319.name	TEXT		StringRange
output.320.name	TEXT		StringRange
output.321.name	TEXT		StringRange
output.322.name	TEXT		StringRange
output.323.name	TEXT		StringRange
output.324.name	TEXT		StringRange
output.325.name	TEXT		StringRange
output.326.name	TEXT		StringRange
output.327.name	TEXT		StringRange
output.328.name	TEXT		StringRange
output.329.name	TEXT		StringRange
output.330.name	TEXT		StringRange
output.331.name	TEXT		StringRange
output.332.name	TEXT		StringRange
output.333.name	TEXT		StringRange
output.334.name	TEXT		StringRange
output.335.name	TEXT		StringRange
output.336.name	TEXT		StringRange
output.337.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.338.name	TEXT		StringRange
output.339.name	TEXT		StringRange
output.340.name	TEXT		StringRange
output.341.name	TEXT		StringRange
output.342.name	TEXT		StringRange
output.343.name	TEXT		StringRange
output.344.name	TEXT		StringRange
output.345.name	TEXT		StringRange
output.346.name	TEXT		StringRange
output.347.name	TEXT		StringRange
output.348.name	TEXT		StringRange
output.349.name	TEXT		StringRange
output.350.name	TEXT		StringRange
output.351.name	TEXT		StringRange
output.352.name	TEXT		StringRange
output.353.name	TEXT		StringRange
output.354.name	TEXT		StringRange
output.355.name	TEXT		StringRange
output.356.name	TEXT		StringRange
output.357.name	TEXT		StringRange
output.358.name	TEXT		StringRange
output.359.name	TEXT		StringRange
output.360.name	TEXT		StringRange
output.361.name	TEXT		StringRange
output.362.name	TEXT		StringRange
output.363.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.364.name	TEXT		StringRange
output.365.name	TEXT		StringRange
output.366.name	TEXT		StringRange
output.367.name	TEXT		StringRange
output.368.name	TEXT		StringRange
output.369.name	TEXT		StringRange
output.370.name	TEXT		StringRange
output.371.name	TEXT		StringRange
output.372.name	TEXT		StringRange
output.373.name	TEXT		StringRange
output.374.name	TEXT		StringRange
output.375.name	TEXT		StringRange
output.376.name	TEXT		StringRange
output.377.name	TEXT		StringRange
output.378.name	TEXT		StringRange
output.379.name	TEXT		StringRange
output.380.name	TEXT		StringRange
output.381.name	TEXT		StringRange
output.382.name	TEXT		StringRange
output.383.name	TEXT		StringRange
output.384.name	TEXT		StringRange
output.385.name	TEXT		StringRange
output.386.name	TEXT		StringRange
output.387.name	TEXT		StringRange
output.388.name	TEXT		StringRange
output.389.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.390.name	TEXT		StringRange
output.391.name	TEXT		StringRange
output.392.name	TEXT		StringRange
output.393.name	TEXT		StringRange
output.394.name	TEXT		StringRange
output.395.name	TEXT		StringRange
output.396.name	TEXT		StringRange
output.397.name	TEXT		StringRange
output.398.name	TEXT		StringRange
output.399.name	TEXT		StringRange
output.400.name	TEXT		StringRange
output.401.name	TEXT		StringRange
output.402.name	TEXT		StringRange
output.403.name	TEXT		StringRange
output.404.name	TEXT		StringRange
output.405.name	TEXT		StringRange
output.406.name	TEXT		StringRange
output.407.name	TEXT		StringRange
output.408.name	TEXT		StringRange
output.409.name	TEXT		StringRange
output.410.name	TEXT		StringRange
output.411.name	TEXT		StringRange
output.412.name	TEXT		StringRange
output.413.name	TEXT		StringRange
output.414.name	TEXT		StringRange
output.415.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.416.name	TEXT		StringRange
output.417.name	TEXT		StringRange
output.418.name	TEXT		StringRange
output.419.name	TEXT		StringRange
output.420.name	TEXT		StringRange
output.421.name	TEXT		StringRange
output.422.name	TEXT		StringRange
output.423.name	TEXT		StringRange
output.424.name	TEXT		StringRange
output.425.name	TEXT		StringRange
output.426.name	TEXT		StringRange
output.427.name	TEXT		StringRange
output.428.name	TEXT		StringRange
output.429.name	TEXT		StringRange
output.430.name	TEXT		StringRange
output.431.name	TEXT		StringRange
output.432.name	TEXT		StringRange
output.433.name	TEXT		StringRange
output.434.name	TEXT		StringRange
output.435.name	TEXT		StringRange
output.436.name	TEXT		StringRange
output.437.name	TEXT		StringRange
output.438.name	TEXT		StringRange
output.439.name	TEXT		StringRange
output.440.name	TEXT		StringRange
output.441.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.442.name	TEXT		StringRange
output.443.name	TEXT		StringRange
output.444.name	TEXT		StringRange
output.445.name	TEXT		StringRange
output.446.name	TEXT		StringRange
output.447.name	TEXT		StringRange
output.448.name	TEXT		StringRange
output.449.name	TEXT		StringRange
output.450.name	TEXT		StringRange
output.451.name	TEXT		StringRange
output.452.name	TEXT		StringRange
output.453.name	TEXT		StringRange
output.454.name	TEXT		StringRange
output.455.name	TEXT		StringRange
output.456.name	TEXT		StringRange
output.457.name	TEXT		StringRange
output.458.name	TEXT		StringRange
output.459.name	TEXT		StringRange
output.460.name	TEXT		StringRange
output.461.name	TEXT		StringRange
output.462.name	TEXT		StringRange
output.463.name	TEXT		StringRange
output.464.name	TEXT		StringRange
output.465.name	TEXT		StringRange
output.466.name	TEXT		StringRange
output.467.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.468.name	TEXT		StringRange
output.469.name	TEXT		StringRange
output.470.name	TEXT		StringRange
output.471.name	TEXT		StringRange
output.472.name	TEXT		StringRange
output.473.name	TEXT		StringRange
output.474.name	TEXT		StringRange
output.475.name	TEXT		StringRange
output.476.name	TEXT		StringRange
output.477.name	TEXT		StringRange
output.478.name	TEXT		StringRange
output.479.name	TEXT		StringRange
output.480.name	TEXT		StringRange
output.481.name	TEXT		StringRange
output.482.name	TEXT		StringRange
output.483.name	TEXT		StringRange
output.484.name	TEXT		StringRange
output.485.name	TEXT		StringRange
output.486.name	TEXT		StringRange
output.487.name	TEXT		StringRange
output.488.name	TEXT		StringRange
output.489.name	TEXT		StringRange
output.490.name	TEXT		StringRange
output.491.name	TEXT		StringRange
output.492.name	TEXT		StringRange
output.493.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.494.name	TEXT		StringRange
output.495.name	TEXT		StringRange
output.496.name	TEXT		StringRange
output.497.name	TEXT		StringRange
output.498.name	TEXT		StringRange
output.499.name	TEXT		StringRange
output.500.name	TEXT		StringRange
output.501.name	TEXT		StringRange
output.502.name	TEXT		StringRange
output.503.name	TEXT		StringRange
output.504.name	TEXT		StringRange
output.505.name	TEXT		StringRange
output.506.name	TEXT		StringRange
output.507.name	TEXT		StringRange
output.508.name	TEXT		StringRange
output.509.name	TEXT		StringRange
output.510.name	TEXT		StringRange
output.511.name	TEXT		StringRange
output.512.name	TEXT		StringRange
input.001.name	TEXT		StringRange
input.002.name	TEXT		StringRange
input.003.name	TEXT		StringRange
input.004.name	TEXT		StringRange
input.005.name	TEXT		StringRange
input.006.name	TEXT		StringRange
input.007.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.008.name	TEXT		StringRange
input.009.name	TEXT		StringRange
input.010.name	TEXT		StringRange
input.011.name	TEXT		StringRange
input.012.name	TEXT		StringRange
input.013.name	TEXT		StringRange
input.014.name	TEXT		StringRange
input.015.name	TEXT		StringRange
input.016.name	TEXT		StringRange
input.017.name	TEXT		StringRange
input.018.name	TEXT		StringRange
input.019.name	TEXT		StringRange
input.020.name	TEXT		StringRange
input.021.name	TEXT		StringRange
input.022.name	TEXT		StringRange
input.023.name	TEXT		StringRange
input.024.name	TEXT		StringRange
input.025.name	TEXT		StringRange
input.026.name	TEXT		StringRange
input.027.name	TEXT		StringRange
input.028.name	TEXT		StringRange
input.029.name	TEXT		StringRange
input.030.name	TEXT		StringRange
input.031.name	TEXT		StringRange
input.032.name	TEXT		StringRange
input.033.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.034.name	TEXT		StringRange
input.035.name	TEXT		StringRange
input.036.name	TEXT		StringRange
input.037.name	TEXT		StringRange
input.038.name	TEXT		StringRange
input.039.name	TEXT		StringRange
input.040.name	TEXT		StringRange
input.041.name	TEXT		StringRange
input.042.name	TEXT		StringRange
input.043.name	TEXT		StringRange
input.044.name	TEXT		StringRange
input.045.name	TEXT		StringRange
input.046.name	TEXT		StringRange
input.047.name	TEXT		StringRange
input.048.name	TEXT		StringRange
input.049.name	TEXT		StringRange
input.050.name	TEXT		StringRange
input.051.name	TEXT		StringRange
input.052.name	TEXT		StringRange
input.053.name	TEXT		StringRange
input.054.name	TEXT		StringRange
input.055.name	TEXT		StringRange
input.056.name	TEXT		StringRange
input.057.name	TEXT		StringRange
input.058.name	TEXT		StringRange
input.059.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.060.name	TEXT		StringRange
input.061.name	TEXT		StringRange
input.062.name	TEXT		StringRange
input.063.name	TEXT		StringRange
input.064.name	TEXT		StringRange
input.065.name	TEXT		StringRange
input.066.name	TEXT		StringRange
input.067.name	TEXT		StringRange
input.068.name	TEXT		StringRange
input.069.name	TEXT		StringRange
input.070.name	TEXT		StringRange
input.071.name	TEXT		StringRange
input.072.name	TEXT		StringRange
input.073.name	TEXT		StringRange
input.074.name	TEXT		StringRange
input.075.name	TEXT		StringRange
input.076.name	TEXT		StringRange
input.077.name	TEXT		StringRange
input.078.name	TEXT		StringRange
input.079.name	TEXT		StringRange
input.080.name	TEXT		StringRange
input.081.name	TEXT		StringRange
input.082.name	TEXT		StringRange
input.083.name	TEXT		StringRange
input.084.name	TEXT		StringRange
input.085.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.086.name	TEXT		StringRange
input.087.name	TEXT		StringRange
input.088.name	TEXT		StringRange
input.089.name	TEXT		StringRange
input.090.name	TEXT		StringRange
input.091.name	TEXT		StringRange
input.092.name	TEXT		StringRange
input.093.name	TEXT		StringRange
input.094.name	TEXT		StringRange
input.095.name	TEXT		StringRange
input.096.name	TEXT		StringRange
input.097.name	TEXT		StringRange
input.098.name	TEXT		StringRange
input.099.name	TEXT		StringRange
input.100.name	TEXT		StringRange
input.101.name	TEXT		StringRange
input.102.name	TEXT		StringRange
input.103.name	TEXT		StringRange
input.104.name	TEXT		StringRange
input.105.name	TEXT		StringRange
input.106.name	TEXT		StringRange
input.107.name	TEXT		StringRange
input.108.name	TEXT		StringRange
input.109.name	TEXT		StringRange
input.110.name	TEXT		StringRange
input.111.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.112.name	TEXT		StringRange
input.113.name	TEXT		StringRange
input.114.name	TEXT		StringRange
input.115.name	TEXT		StringRange
input.116.name	TEXT		StringRange
input.117.name	TEXT		StringRange
input.118.name	TEXT		StringRange
input.119.name	TEXT		StringRange
input.120.name	TEXT		StringRange
input.121.name	TEXT		StringRange
input.122.name	TEXT		StringRange
input.123.name	TEXT		StringRange
input.124.name	TEXT		StringRange
input.125.name	TEXT		StringRange
input.126.name	TEXT		StringRange
input.127.name	TEXT		StringRange
input.128.name	TEXT		StringRange
output.001	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.002	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.003	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.004	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.005	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.006	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.007	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.008	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.009	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.010	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.011	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.012	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.013	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.014	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.015	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.016	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.017	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.018	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.019	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.020	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.021	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.022	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.023	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.024	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.025	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.026	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.027	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.028	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.029	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.030	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.031	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.032	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.033	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.034	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.035	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.036	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.037	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.038	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.039	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.040	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.041	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.042	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.043	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.044	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.045	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.046	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.047	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.048	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.049	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.050	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.051	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.052	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.053	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.054	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.055	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.056	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.057	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.058	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.059	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.060	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.061	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.062	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.063	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.064	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.065	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.066	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.067	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.068	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.069	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.070	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.071	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.072	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.073	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.074	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.075	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.076	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.01	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.077	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.078	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.079	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.080	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.081	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.082	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.083	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.084	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.085	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.086	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.087	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.088	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.089	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.090	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.091	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.092	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.093	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.094	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.095	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.096	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.097	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.098	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.099	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.100	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.101	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.102	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.103	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.104	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.105	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.106	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.107	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.108	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.109	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.110	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.111	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.112	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.113	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.114	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.115	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.116	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.117	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.118	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.119	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.120	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.121	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.122	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.123	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.124	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.125	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.126	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.127	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.128	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.129	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.130	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.131	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.132	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.133	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.134	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.135	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.136	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.137	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.138	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.139	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.140	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.141	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.142	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.143	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.144	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.145	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.146	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.147	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.148	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.149	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.150	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.151	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.152	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.153	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.154	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.155	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.156	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.157	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.158	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.159	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.160	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.161	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.162	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.163	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.164	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.165	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.166	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.167	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.168	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.169	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.170	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.171	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.172	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.173	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.174	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.175	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.176	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.177	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.178	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.179	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.180	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.181	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.182	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.183	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.184	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.185	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.186	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.187	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.188	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.189	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.190	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.191	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.192	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.193	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.194	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.195	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.196	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.197	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.198	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.199	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.200	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.201	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.202	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.203	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.204	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.205	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.206	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.207	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.208	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.209	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.210	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.211	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.212	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.213	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.214	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.215	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.216	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.217	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.218	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.219	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.220	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.221	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.222	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.223	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.224	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.225	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.226	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.227	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.228	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.229	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.230	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.231	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.232	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.233	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.234	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.235	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.236	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.237	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.238	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.239	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.240	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.241	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.242	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.243	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.244	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.245	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.246	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.247	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.248	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.249	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.250	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.251	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.252	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.253	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.254	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.255	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.256	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.257	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.258	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.259	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.260	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.261	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.262	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.263	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.264	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.265	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.266	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.267	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.268	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.269	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.270	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.271	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.272	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.273	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.274	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.275	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.276	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.277	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.278	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.279	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.280	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.281	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.282	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.283	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.284	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.285	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.286	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.287	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.288	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.289	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.290	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.291	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.292	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.293	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.294	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.295	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.296	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.297	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.298	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.299	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.300	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.301	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.302	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.303	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.304	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.305	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.306	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.307	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.308	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.309	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.310	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.311	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.312	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.313	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.314	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.315	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.316	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.317	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.318	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.319	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.320	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.321	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.322	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.323	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.324	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.325	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.326	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.327	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.328	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.329	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.330	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.331	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.332	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.333	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.334	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.335	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.336	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.337	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.338	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.339	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.340	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.341	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.342	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.343	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.344	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.345	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.346	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.347	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.348	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.349	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.350	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.351	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.352	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.353	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.354	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.355	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.356	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.357	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.358	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.359	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.360	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.361	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.362	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.363	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.364	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.365	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.366	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.367	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.368	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.369	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.370	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.371	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.372	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.373	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.374	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.375	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.376	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.377	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.378	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.379	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.380	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.381	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.382	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.383	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.384	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.385	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.386	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.387	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.388	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.389	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.390	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.391	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.392	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.393	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.394	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.395	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.396	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.397	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.398	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.399	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.400	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.401	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.402	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.403	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.404	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.405	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.406	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.407	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.408	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.409	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.410	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.411	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.412	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.413	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.414	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.415	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.416	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.417	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.418	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.419	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.420	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.421	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.422	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.423	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.424	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.425	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.426	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.427	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.428	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.429	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.430	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.431	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.432	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.433	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.434	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.435	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.436	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.437	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.438	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.439	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.440	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.441	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.442	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.443	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.444	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.445	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.446	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.447	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.448	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.449	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.450	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.451	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.452	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.453	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.454	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.455	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.456	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.457	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.458	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.459	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.460	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.461	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.462	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.463	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.464	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.465	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.466	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.467	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.468	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.469	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.470	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.471	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.472	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.473	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.474	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.475	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.476	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.477	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.478	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.479	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.480	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.481	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.482	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.483	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.484	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.485	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.486	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.487	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.488	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.489	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.490	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.491	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.492	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.493	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.494	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.495	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.496	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.497	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.498	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.499	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.500	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.501	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.502	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.503	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.504	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.505	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.506	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.507	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.508	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.509	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.510	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
output.511	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
output.512	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.001	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.002	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.003	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.004	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.005	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.006	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.007	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.008	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.009	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.010	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.011	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.012	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.013	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.014	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.015	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.016	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.017	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.018	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.019	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.020	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.021	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.022	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.023	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.024	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.025	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.026	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.027	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.028	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.029	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.030	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.031	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.032	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.033	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.034	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.035	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.036	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.037	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.038	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.039	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.040	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.041	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.042	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.043	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.044	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.045	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.046	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.047	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.048	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.049	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.050	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.051	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.052	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.053	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.054	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.055	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.056	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.057	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.058	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.059	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.060	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.061	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.062	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.063	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.064	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.065	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.066	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.067	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.068	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.069	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.070	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.071	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.072	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.073	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.074	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.075	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.076	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.077	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.078	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.079	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.080	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.081	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.082	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.083	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.084	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.085	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.086	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.087	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.088	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.089	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.090	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.091	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.092	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.093	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.094	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.095	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.096	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.097	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.098	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.099	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.100	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.101	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.102	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.103	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.104	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.105	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.106	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.107	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.108	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.109	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.110	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.111	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.112	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.113	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.114	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.115	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.116	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.117	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.118	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.119	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.120	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.121	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.122	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.123	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.124	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.125	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.126	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.127	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.128	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.129	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.130	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.131	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.132	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.133	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.134	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.135	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.136	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.137	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.138	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.139	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.140	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.141	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.142	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.143	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.144	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.145	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.146	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.147	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.148	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.149	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.150	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.151	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.152	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.153	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.154	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.155	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.156	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.157	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.158	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.159	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.160	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.161	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.162	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.163	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.164	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.165	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.166	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.167	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.168	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.169	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.170	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.171	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.172	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.173	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.174	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.175	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.176	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.177	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.178	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.179	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.180	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.181	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.182	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.183	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.184	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.185	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.186	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.187	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.188	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.189	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.190	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.191	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.192	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.193	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.194	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.195	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.196	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.197	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.198	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.199	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.200	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.201	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.202	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.203	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.204	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.205	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.206	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.207	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.208	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.209	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.210	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.211	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.212	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.213	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.214	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.215	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.216	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.217	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.218	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.219	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.220	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.221	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.222	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.223	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.224	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.225	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.226	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.227	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.228	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.229	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.230	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.231	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.232	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.233	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.234	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.235	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.236	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.237	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.238	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.239	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.240	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.241	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.242	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.243	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.244	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.245	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.246	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.247	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.248	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.249	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.250	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.251	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.252	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.253	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.254	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.255	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.256	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.257	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.258	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.259	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.260	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.261	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.262	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.263	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.264	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.265	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.266	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.267	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.268	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.269	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.270	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.271	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.272	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.273	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.274	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.275	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.276	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.277	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.278	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.279	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.280	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.281	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.282	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.283	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.284	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.285	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.286	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.287	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.288	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.289	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.290	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.291	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.292	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.293	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.294	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.295	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.296	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.297	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.298	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.299	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.300	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.301	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.302	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.303	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.304	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.305	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.306	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.307	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.308	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.309	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.310	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.311	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.312	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.313	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.314	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.315	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.316	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.317	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.318	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.319	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.320	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.321	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.322	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.323	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.324	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.325	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.326	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.327	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.328	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.329	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.330	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.331	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.332	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.333	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.334	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.335	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.336	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.337	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.338	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.339	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.340	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.341	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.342	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.343	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.344	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.345	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.346	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.347	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.348	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.349	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.350	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.351	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.352	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.353	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.354	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.355	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.356	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.357	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.358	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.359	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.360	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.361	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.362	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.363	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.364	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.365	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.366	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.367	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.368	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.369	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.370	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.371	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.372	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.373	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.374	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.375	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.376	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.377	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.378	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.379	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.380	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.381	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.382	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.383	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.384	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.385	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.386	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.387	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.388	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.389	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.390	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.391	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.392	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.393	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.394	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.395	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.396	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.397	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.398	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.399	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.400	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.401	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.402	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.403	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.404	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.405	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.406	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.407	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.408	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.409	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.410	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.411	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.412	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.413	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.414	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.415	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.416	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.417	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.418	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.419	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.420	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.421	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.422	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.423	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.424	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.425	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.426	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.427	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.428	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.429	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.430	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.431	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.432	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.433	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.434	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.435	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.436	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.437	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.438	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.439	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.440	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.441	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.442	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.443	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.444	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.445	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.446	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.447	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.448	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.449	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.450	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.451	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.452	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.453	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.454	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.455	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.456	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.457	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.458	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.459	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.460	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.461	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.462	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.463	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.464	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.465	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.466	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.467	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.468	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.469	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.470	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.471	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.472	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.473	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.474	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.475	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.476	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.477	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.478	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.479	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.480	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.481	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.482	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.483	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.484	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.485	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.486	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.487	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.488	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.489	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.490	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.491	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.492	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.493	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.494	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.495	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.496	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.497	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.498	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.499	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.500	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.501	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.502	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.503	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.504	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.505	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.506	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.507	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.508	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.509	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.510	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
byname.511	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)

name	*type*	*flags*	*range*
byname.512	CHOICE		EnumRange (OFF 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128)
state.001.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.001.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.002.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.002.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.003.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.003.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.004.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.004.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.005.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.005.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.006.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.006.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.007.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.007.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.008.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.008.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.009.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.009.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.010.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.010.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.011.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.011.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.012.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.012.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.013.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.013.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.014.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.014.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.015.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.015.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.016.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.016.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.017.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.017.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.018.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.018.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.019.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.019.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.020.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.020.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.021.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.021.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.022.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.022.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.023.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.023.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.024.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.024.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.025.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.025.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.026.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.026.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.027.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.027.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.028.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.028.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.029.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.029.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.030.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.030.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.031.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.031.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.032.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.032.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.033.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.033.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.034.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.034.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.035.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.035.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.036.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.036.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.037.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.037.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.038.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.038.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.039.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.039.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.040.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.040.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.041.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.041.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.042.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.042.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.043.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.043.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.044.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.044.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.045.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.045.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.046.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.046.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.047.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.047.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.048.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.048.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.049.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.049.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.050.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.050.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.051.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.051.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.052.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.052.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.053.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.053.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.054.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.054.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.055.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.055.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.056.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.056.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.057.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.057.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.058.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.058.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.059.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.059.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.060.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.060.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.061.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.061.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.062.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.062.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.063.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.063.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.064.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.064.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.065.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.065.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.066.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.066.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.067.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.067.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.068.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.068.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.069.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.069.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.070.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.070.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.071.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.071.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.072.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.072.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.073.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.073.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.074.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.074.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.075.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.075.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.076.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.076.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.077.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.077.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.078.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.078.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.079.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.079.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.080.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.080.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.081.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.081.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.082.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.082.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.083.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.083.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.084.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.084.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.085.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.085.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.086.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.086.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.087.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.087.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.088.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.088.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.089.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.089.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.090.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.090.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.091.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.091.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.092.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.092.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.093.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.093.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.094.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.094.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.095.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.095.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.096.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.096.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.097.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.097.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.098.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.098.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.099.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.099.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.100.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.100.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.101.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.101.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.102.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.102.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.103.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.103.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.104.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.104.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.105.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.105.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.106.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.106.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.107.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.107.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.108.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.108.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.109.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.109.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.110.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.110.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.111.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.111.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.112.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.112.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.113.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.113.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.114.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.114.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.115.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.115.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.116.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.116.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.117.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.117.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.118.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.118.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.119.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.119.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.120.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.120.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.121.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.121.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.122.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.122.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.123.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.123.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.124.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.124.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.125.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.125.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.126.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.126.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.127.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.127.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
state.128.rfLow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.128.rfHigh	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 1 PSU 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 1 PSU 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Module 2 PSU 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Module 2 PSU 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Module 3 PSU 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Module 3 PSU 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module 4 PSU 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module 4 PSU 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 4)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 9)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 12)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 13)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 14)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 15)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Summary IMC 16)

9.225 Emerson-AC-HPM

This page describes the device driver and the device window for the Emerson / Liebert Air Conditioning System Model HPM with IntelliSlot Web Card.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.agentModel	TEXT	R/O	StringRange R/O
info.agentFirmware	TEXT	R/O	StringRange R/O
info.agentSerial	TEXT	R/O	StringRange R/O
info.deviceModel	TEXT	R/O	StringRange R/O
info.deviceName	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.maintenanceMargin	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
operationState	CHOICE	R/O	EnumRange R/O (NORMAL STARTUP NORMAL+WARNING NORMAL+ALARM ABNORMAL)
systemState	CHOICE	R/O	EnumRange R/O (OFF ON STANDBY)
operationMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
meas.temp.supplyAir	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.returnAir	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.humidity.returnAir	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.humidifying	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.humidifying	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dehumidifying	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.dehumidifying	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fan	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.fan	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.heating	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.heating	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.freeCooling	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.freeCooling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.cooling	CHOICE	R/O	EnumRange R/O (OFF ON)
meas.cooling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
maintenance.capacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
maintenance.year	TEXT	NOPRESET	StringRange
maintenance.month	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AC Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Maintenance)

9.226 Endrun-FDC3300

This page describes the device driver and the device window for the EndRun 2x10 Frequency Distribution Unit FDC-3300.

The driver was tested with the following models:

- FDC-3300, Version 6010

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device and for proper operation.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
externalAlarm	Defines the usage of the two disable inputs and the disablemode behaviour of the FDC unit. If disablemode=y,y,off (or n,y,off or y,n,off) the chassis will turn off the reference input when it sees the disable input asserted and there is no good signal present on the other signal input. If disablemode=y,y,on (or n,y,on or y,n,on) then the chassis will switch between the primary and secondary inputs when the disable input is asserted.

Remote Control

- Default serial port parameters RS232,19200,8,None,1
- Default TCP-Port 4001

- Please set respmode = terse

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.versionFpga	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.externalAlarm	CHOICE	SETUP	EnumRange (Y-Y-ON Y-Y-OFF Y-N-ON Y-N-OFF N-Y-ON N-Y-OFF N-N)
position	CHOICE	R/O	EnumRange R/O (A B)
reset	TEXT	NOPRESET	StringRange
switchmode	CHOICE		EnumRange (A ONLY B ONLY A PRIMARY B PRIMARY)
internal.extAlarm	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 4)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power Supply A)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power Supply B)
faults.15	ALARM	R/O	AlarmFlagRange R/O (System oscillator)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Flash)
faults.17	ALARM	R/O	AlarmFlagRange R/O (FPGA)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Disable External A)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Disable External B)

9.227 Enensys-Vyper-Modulator

This page describes the device driver and the device window for the Enensys Vyper Modulator over SNMP.

It was tested with Model XSSR-VYP0-3001

- Firmware version: 2.17 HW 106

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
bucVoltage	Sets the BUC voltage on/off. Only if BUC dc supply is supported by the modulator
unmuteType	Sets the unmute type of the modulator
muteOnInputError	Sets the mute on input error behaviour
muteAtStartup	Sets the mute at startup behaviour
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
info.redUnit	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.bucVoltage	CHOICE	SETUP	EnumRange (OFF ON)
config.unmuteType	CHOICE	SETUP	EnumRange (DIRECT PROGRESSIVE)
config.muteOnInputErr	CHOICE	SETUP	EnumRange (ON MUTE)

name	*type*	*flags*	*range*
config.muteAtStartup	CHOICE	SETUP	EnumRange (ON MUTE)
config.redundancy	CHOICE	SETUP	EnumRange (OFF ON)
input.single.main	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.single.backup1	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.single.backup2	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.single.switch	CHOICE		EnumRange (OFF ON)
input.mts.1.main	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.1.backup	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.1.switch	CHOICE		EnumRange (OFF ON)
input.mts.2.main	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.2.backup	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.2.switch	CHOICE		EnumRange (OFF ON)
input.mts.3.main	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.3.backup	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.3.switch	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
input.mts.4.main	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.4.backup	CHOICE		EnumRange (ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 NONE)
input.mts.4.switch	CHOICE		EnumRange (OFF ON)
input.tsSync	CHOICE		EnumRange (OFF ON)
input.bitrateOverflow	CHOICE		EnumRange (OFF ON)
input.bitrateRatio	CHOICE		EnumRange (OFF ON)
input.jitterDegrad	CHOICE		EnumRange (OFF ON)
input.currentJitter	CHOICE		EnumRange (OFF ON)
input.bitrateRatioThd	FLOAT		DoubleRange (1.0 .. 100.0)
input.switchDelay	INTEGER		IntegerRange (0 .. 10)
input.jitterDegradThd	FLOAT		DoubleRange (1.0 .. 200.0)
input.currentJitterThd	FLOAT		DoubleRange (0.000008 .. 1000.000000)
rfBand	CHOICE		EnumRange (IF-BAND L-BAND)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.power	FLOAT		DoubleRange (-35.0 .. 7.0)
tx.mod.refOutput	CHOICE		EnumRange (OFF ON)
tx.mod.refLevel	INTEGER		IntegerRange (-3 .. 3)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.cwMode	CHOICE		EnumRange (MOD CW)
tx.custom.rollOff	INTEGER		IntegerRange (5 .. 35)
tx.custom.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.input	CHOICE		EnumRange (OFF ASI1 ASI2 ASI3 ASI4 TSoIP1 TSoIP2 TSoIP3 TSoIP4 ActiveStream)
tx.mod.standard	CHOICE		EnumRange (DVB-S DVB-S2 DVB-S2X)
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.testMode	CHOICE		EnumRange (OFF PBRS DUMMY PL FRAMES CARRIER ID ONLY)
tx.mod.rollOff	CHOICE		EnumRange (35% 25% 20% 15% 10% 5%)
tx.mod.acmMode	CHOICE		EnumRange (ACM-VCM CCM)
tx.mod.plScrambling1	INTEGER		IntegerRange (0 .. 262141)
tx.mod.plScrambling2	INTEGER		IntegerRange (0 .. 262141)
tx.mod.plScrambling	CHOICE		EnumRange (DEFAULT PL SEQUENCE 1 PL SEQUENCE 2)
tx.mod.rateAdaptation	CHOICE		EnumRange (ADV. PADDING INSERT NULL PACKET ADAPT PADDING INSERT DUMMY PL FRAMES)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)
tx.mod.frameSize	CHOICE		EnumRange (LONG SHORT)
tx.mod.streamType	CHOICE		EnumRange (SINGLE MULTI)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mod.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mod.issyLength	CHOICE		EnumRange (SHORT LONG)
tx.mod.modcod	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-7/8 8PSK-2/3 8PSK-5/6 8PSK-8/9 16QAM-3/4 16QAM-7/8)

name	*type*	*flags*	*range*
tx.mod.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK VLSNR-GPSK VLSNR-pi2-BPSK N/A)
tx.mod.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.1.enable	CHOICE		EnumRange (OFF ON)
tx.mts.2.enable	CHOICE		EnumRange (OFF ON)
tx.mts.3.enable	CHOICE		EnumRange (OFF ON)
tx.mts.4.enable	CHOICE		EnumRange (OFF ON)
tx.mts.1.isi	INTEGER		IntegerRange (1 .. 255)
tx.mts.2.isi	INTEGER		IntegerRange (1 .. 255)
tx.mts.3.isi	INTEGER		IntegerRange (1 .. 255)
tx.mts.4.isi	INTEGER		IntegerRange (1 .. 255)
tx.mts.1.modcod	CHOICE		EnumRange (DUMMY-PL-FRAME QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)

name	*type*	*flags*	*range*
tx.mts.2.modcod	CHOICE		EnumRange (DUMMY-PL-FRAME QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mts.3.modcod	CHOICE		EnumRange (DUMMY-PL-FRAME QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mts.4.modcod	CHOICE		EnumRange (DUMMY-PL-FRAME QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)

name	*type*	*flags*	*range*
tx.mts.1.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.2.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.3.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.4.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.1.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK VLSNR-GPSK VLSNR-pi2-BPSK N/A)
tx.mts.2.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK VLSNR-GPSK VLSNR-pi2-BPSK N/A)

name	*type*	*flags*	*range*
tx.mts.3.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK VLSNR-GPSK VLSNR-pi2-BPSK N/A)
tx.mts.4.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK VLSNR-GPSK VLSNR-pi2-BPSK N/A)
tx.mts.1.frameSize	CHOICE		EnumRange (LONG SHORT)
tx.mts.2.frameSize	CHOICE		EnumRange (LONG SHORT)
tx.mts.3.frameSize	CHOICE		EnumRange (LONG SHORT)
tx.mts.4.frameSize	CHOICE		EnumRange (LONG SHORT)
tx.mts.1.pilots	CHOICE		EnumRange (OFF ON)
tx.mts.2.pilots	CHOICE		EnumRange (OFF ON)
tx.mts.3.pilots	CHOICE		EnumRange (OFF ON)
tx.mts.4.pilots	CHOICE		EnumRange (OFF ON)
tx.mts.1.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.2.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.3.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.4.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.1.issyLength	CHOICE		EnumRange (SHORT LONG)
tx.mts.2.issyLength	CHOICE		EnumRange (SHORT LONG)
tx.mts.3.issyLength	CHOICE		EnumRange (SHORT LONG)
tx.mts.4.issyLength	CHOICE		EnumRange (SHORT LONG)
tx.mts.1.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mts.2.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mts.3.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mts.4.nullPacketDelete	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mts.1.bitrateLimit	FLOAT		DoubleRange (1.000000 .. 500.000000)
tx.mts.2.bitrateLimit	FLOAT		DoubleRange (1.000000 .. 500.000000)
tx.mts.3.bitrateLimit	FLOAT		DoubleRange (1.000000 .. 500.000000)
tx.mts.4.bitrateLimit	FLOAT		DoubleRange (1.000000 .. 500.000000)
tx.mts.1.prbs	CHOICE		EnumRange (OFF ON)
tx.mts.2.prbs	CHOICE		EnumRange (OFF ON)
tx.mts.3.prbs	CHOICE		EnumRange (OFF ON)
tx.mts.4.prbs	CHOICE		EnumRange (OFF ON)
refClkSrc	CHOICE		EnumRange (INTERN EXTERN)
refClkTimeout	INTEGER		IntegerRange (0 .. 1440)
refClkMute	CHOICE		EnumRange (OFF ON)
refClkDirection	CHOICE		EnumRange (INPUT OUTPUT)
refClkType	CHOICE		EnumRange (10MHz PPS)
ipin.igmpVersion	CHOICE		EnumRange (v2 v3)
ipin.1.port	INTEGER		IntegerRange (0 .. 65535)
ipin.1.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange
ipin.1.latency	INTEGER		IntegerRange (10 .. 3500)
ipin.1.rxTimeout	INTEGER		IntegerRange (0 .. 120)
ipin.1.smoothing	CHOICE		EnumRange (OFF ON)
ipin.1.interface	CHOICE		EnumRange (ETH-1 ETH-2)
ipin.1.redundancy	CHOICE		EnumRange (OFF ON)
ipin.1.redTimeout	INTEGER		IntegerRange (1 .. 10)
ipin.1.manualToggle	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
ipin.1.igmpMode	CHOICE		EnumRange (EXCLUDE INCLUDE)
ipin.1.sourceFilter	CHOICE		EnumRange (OFF ON)
ipin.1.sourceAddress	TEXT		StringRange
ipin.1.sourcePort	INTEGER		IntegerRange (0 .. 65535)
ipin.2.port	INTEGER		IntegerRange (0 .. 65535)
ipin.2.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange
ipin.2.latency	INTEGER		IntegerRange (10 .. 3500)
ipin.2.rxTimeout	INTEGER		IntegerRange (0 .. 120)
ipin.2.smoothing	CHOICE		EnumRange (OFF ON)
ipin.2.interface	CHOICE		EnumRange (ETH-1 ETH-2)
ipin.2.redundancy	CHOICE		EnumRange (OFF ON)
ipin.2.redTimeout	INTEGER		IntegerRange (1 .. 10)
ipin.2.manualToggle	TEXT	NOPRESET	StringRange
ipin.2.igmpMode	CHOICE		EnumRange (EXCLUDE INCLUDE)
ipin.2.sourceFilter	CHOICE		EnumRange (OFF ON)
ipin.2.sourceAddress	TEXT		StringRange
ipin.2.sourcePort	INTEGER		IntegerRange (0 .. 65535)
ipin.3.port	INTEGER		IntegerRange (0 .. 65535)
ipin.3.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.3.address	TEXT		StringRange
ipin.3.latency	INTEGER		IntegerRange (10 .. 3500)
ipin.3.rxTimeout	INTEGER		IntegerRange (0 .. 120)
ipin.3.smoothing	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
ipin.3.interface	CHOICE		EnumRange (ETH-1 ETH-2)
ipin.3.redundancy	CHOICE		EnumRange (OFF ON)
ipin.3.redTimeout	INTEGER		IntegerRange (1 .. 10)
ipin.3.manualToggle	TEXT	NOPRESET	StringRange
ipin.3.igmpMode	CHOICE		EnumRange (EXCLUDE INCLUDE)
ipin.3.sourceFilter	CHOICE		EnumRange (OFF ON)
ipin.3.sourceAddress	TEXT		StringRange
ipin.3.sourcePort	INTEGER		IntegerRange (0 .. 65535)
ipin.4.port	INTEGER		IntegerRange (0 .. 65535)
ipin.4.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.4.address	TEXT		StringRange
ipin.4.latency	INTEGER		IntegerRange (10 .. 3500)
ipin.4.rxTimeout	INTEGER		IntegerRange (0 .. 120)
ipin.4.smoothing	CHOICE		EnumRange (OFF ON)
ipin.4.interface	CHOICE		EnumRange (ETH-1 ETH-2)
ipin.4.redundancy	CHOICE		EnumRange (OFF ON)
ipin.4.redTimeout	INTEGER		IntegerRange (1 .. 10)
ipin.4.manualToggle	TEXT	NOPRESET	StringRange
ipin.4.igmpMode	CHOICE		EnumRange (EXCLUDE INCLUDE)
ipin.4.sourceFilter	CHOICE		EnumRange (OFF ON)
ipin.4.sourceAddress	TEXT		StringRange
ipin.4.sourcePort	INTEGER		IntegerRange (0 .. 65535)
tx.mux.cid.enable	CHOICE	NOPRESET	EnumRange (OFF ON)
tx.mux.cid.carrierId	TEXT	R/O	StringRange R/O
tx.mux.cid.telnNo	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
tx.mux.cid.longitude	TEXT	NOPRESET	StringRange
tx.mux.cid.userInfo	TEXT	NOPRESET	StringRange
tx.mux.cid.latitude	TEXT	NOPRESET	StringRange
tx.mux.cid.gpsMode	CHOICE		EnumRange (MANUAL GPS)
active.input	TEXT	R/O	StringRange R/O
flags.active	BOOLEAN	R/O	BooleanRange R/O (true false)
input.single.active	CHOICE	R/O	EnumRange R/O (A B)
active.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
active.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
flags.asi1	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.1.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.1.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.1.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.1.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.asi2	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.2.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.2.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.2.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.2.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.asi3	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.3.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
asiin.3.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.3.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.3.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.asi4	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.4.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.4.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asiin.4.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.4.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.ipin1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rtp1	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.1.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
ipin.1.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.ipin2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rtp2	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.2.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
ipin.2.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.ipin3	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rtp3	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ipin.3.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
ipin.3.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
flags.ipin4	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rtp4	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.4.datarateTotal	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.datarateUseful	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
ipin.4.datarateRatio	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.refClkSrc	TEXT	R/O	StringRange R/O
flags.refClkExt	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.pllLock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ppsLock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.refClkType	TEXT	R/O	StringRange R/O
state.ppsMissing	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.rf.on	TEXT	R/O	StringRange R/O
state.tx.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.testMode	TEXT	R/O	StringRange R/O
state.esNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.spectralEff	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.1.activeInput	TEXT	R/O	StringRange R/O
flags.mts.1.fifoFlow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.mts.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.1.bandwidthUseful	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.1.bandwidthTotal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.1.esNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.1.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.1.datarateMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.2.activeInput	TEXT	R/O	StringRange R/O
flags.mts.2.fifoFlow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.mts.2.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.2.bandwidthUseful	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.2.bandwidthTotal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.2.esNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.2.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.2.datarateMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.3.activeInput	TEXT	R/O	StringRange R/O
flags.mts.3.fifoFlow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.mts.3.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.3.bandwidthUseful	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.3.bandwidthTotal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.mts.3.esNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.3.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.3.datarateMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.4.activeInput	TEXT	R/O	StringRange R/O
flags.mts.4.fifoFlow	BOOLEAN	R/O	BooleanRange R/O (true false)
state.mts.4.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.mts.4.bandwidthUseful	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.4.bandwidthTotal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.4.esNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.4.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mts.4.datarateMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.protocol	TEXT	R/O	StringRange R/O
ipin.1.meas.packetSize	TEXT	R/O	StringRange R/O
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecMode	TEXT	R/O	StringRange R/O
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fer	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.1.packetsUncorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.meas.smoothing	TEXT	R/O	StringRange R/O
ipin.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
ipin.1.jitterSupported	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.meas.latency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.1.packetGap	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.jitter	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.jitterMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.jitterDegrad	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.1.jitterAvail	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.1.vlan	TEXT	R/O	StringRange R/O
ipin.1.meas.address	TEXT	R/O	StringRange R/O
ipin.1.packetRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.redUnlockTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.meas.interface	TEXT	R/O	StringRange R/O
ipin.2.protocol	TEXT	R/O	StringRange R/O
ipin.2.meas.packetSize	TEXT	R/O	StringRange R/O
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecMode	TEXT	R/O	StringRange R/O
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fer	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.2.packetsUncorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.meas.smoothing	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.2.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.jitterSupported	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.meas.latency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.2.packetGap	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.jitter	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.jitterMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.jitterDegrad	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.2.jitterAvail	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.2.vlan	TEXT	R/O	StringRange R/O
ipin.2.meas.address	TEXT	R/O	StringRange R/O
ipin.2.packetRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.redUnlockTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.meas.interface	TEXT	R/O	StringRange R/O
ipin.3.protocol	TEXT	R/O	StringRange R/O
ipin.3.meas.packetSize	TEXT	R/O	StringRange R/O
ipin.3.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.fecMode	TEXT	R/O	StringRange R/O
ipin.3.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.fer	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.3.packetsUncorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.3.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.meas.smoothing	TEXT	R/O	StringRange R/O
ipin.3.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.jitterSupported	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.meas.latency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.3.packetGap	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.jitter	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.jitterMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.3.jitterDegrad	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.3.jitterAvail	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.3.vlan	TEXT	R/O	StringRange R/O
ipin.3.meas.address	TEXT	R/O	StringRange R/O
ipin.3.packetRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.redUnlockTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.3.meas.interface	TEXT	R/O	StringRange R/O
ipin.4.protocol	TEXT	R/O	StringRange R/O
ipin.4.meas.packetSize	TEXT	R/O	StringRange R/O
ipin.4.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.fecMode	TEXT	R/O	StringRange R/O
ipin.4.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.4.fer	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.4.packetsUncorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.meas.smoothing	TEXT	R/O	StringRange R/O
ipin.4.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.jitterSupported	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.meas.latency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.4.packetGap	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.jitter	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.jitterMax	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.4.jitterDegrad	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.4.jitterAvail	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.4.vlan	TEXT	R/O	StringRange R/O
ipin.4.meas.address	TEXT	R/O	StringRange R/O
ipin.4.packetRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.redUnlockTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.4.meas.interface	TEXT	R/O	StringRange R/O
ipin.1.reset	TEXT	NOPRESET	StringRange
ipin.1.clearJitter	TEXT	NOPRESET	StringRange
ipin.2.reset	TEXT	NOPRESET	StringRange
ipin.2.clearJitter	TEXT	NOPRESET	StringRange
ipin.3.reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
ipin.3.clearJitter	TEXT	NOPRESET	StringRange
ipin.4.reset	TEXT	NOPRESET	StringRange
ipin.4.clearJitter	TEXT	NOPRESET	StringRange
reboot	TEXT	NOPRESET	StringRange
clearWarning	TEXT	NOPRESET	StringRange
clearCritical	TEXT	NOPRESET	StringRange
unit.protection	CHOICE		EnumRange (MANUAL AUTO)
unit.position	CHOICE		EnumRange (MOD-A MOD-B)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)
position	CHOICE		EnumRange DIS (A B)
location	CHOICE	R/O	EnumRange R/O DIS (ONLINE STANDBY)
internal.input	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (HW Status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (SW Status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Loading Proc)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal Com)
faults.05	ALARM	R/O	AlarmFlagRange R/O (SW Compatibility)
faults.06	ALARM	R/O	AlarmFlagRange R/O (System Clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Startup)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Active Lock)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Active Overflow)
faults.11	ALARM	R/O	AlarmFlagRange R/O (ASI 1 Lock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (ASI 2 Lock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (ASI 3 Lock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ASI 4 Lock)
faults.15	ALARM	R/O	AlarmFlagRange R/O (ASI 1 Overflow)
faults.16	ALARM	R/O	AlarmFlagRange R/O (ASI 2 Overflow)
faults.17	ALARM	R/O	AlarmFlagRange R/O (ASI 3 Overflow)
faults.18	ALARM	R/O	AlarmFlagRange R/O (ASI 4 Overflow)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TSoIP 1 Lock)
faults.22	ALARM	R/O	AlarmFlagRange R/O (TSoIP 2 Lock)
faults.23	ALARM	R/O	AlarmFlagRange R/O (TSoIP 3 Lock)
faults.24	ALARM	R/O	AlarmFlagRange R/O (TSoIP 4 Lock)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TSoIP 1 RTP Lock)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TSoIP 2 RTP Lock)
faults.27	ALARM	R/O	AlarmFlagRange R/O (TSoIP 3 RTP Lock)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (TSoIP 4 RTP Lock)
faults.29	ALARM	R/O	AlarmFlagRange R/O (TSoIP 1 Overflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (TSoIP 2 Overflow)
faults.31	ALARM	R/O	AlarmFlagRange R/O (TSoIP 3 Overflow)
faults.32	ALARM	R/O	AlarmFlagRange R/O (TSoIP 4 Overflow)
faults.41	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Fan 5)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Warning Relay)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Critical Relay)

9.228 Envivio-4Caster-B3

This page describes the device driver and the device window for the Envivio 4Caster B3 MPEG-4 video encoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- IP and MPEG output parameters
- --- Video parameters
- --- Audio parameters

- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The only configuration parameter which must be set for the Envivio encoder is it's IP address:

parameter	***description***
address	Enter the IP address of the encoder here. Example:
10.0.0.1	

Remarks

- To change parameters at the device, the encoder must be in stopped mode. The changes will be applied when the encoder is started again.
- Loading and saving presets for this device is only possible if the device is in STOPPED mode.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.systemVersion	TEXT	R/O	StringRange R/O
info.4casterVersion	TEXT	R/O	StringRange R/O
info.h264Encoder	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.aspEncoder	TEXT	R/O	StringRange R/O
info.aacEncoder	TEXT	R/O	StringRange R/O
info.avInputDriver	TEXT	R/O	StringRange R/O
input.video.frameRate	INTEGER		IntegerRange (0 .. 0)
input.video.width	INTEGER		IntegerRange (0 .. 0)
input.video.height	INTEGER		IntegerRange (0 .. 0)
input.video.useDeinterlaceFilter	CHOICE		EnumRange (OFF ON)
input.video.brightness	INTEGER		IntegerRange (0 .. 100)
input.video.contrast	INTEGER		IntegerRange (0 .. 100)
input.video.saturation	INTEGER		IntegerRange (0 .. 100)
input.video.type	CHOICE		EnumRange (SD Composite Component S-Video)
input.video.enabled	CHOICE		EnumRange (OFF ON)
input.audio.frequency	CHOICE		EnumRange (AUTO 48 kHz 32 kHz 24 kHz 16 kHz 12 kHz 8 kHz)
input.audio.numChannels	CHOICE		EnumRange (LEFT RIGHT DUAL STEREO)
input.audio.volume	INTEGER		IntegerRange (0 .. 100)

name	*type*	*flags*	*range*
input.audio.languageCode	CHOICE		EnumRange (None English French Spanish German)
input.audio.type	CHOICE		EnumRange (None SDI (G1 P1) SDI (G1 P2) SDI (G2 P1) SDI (G2 P2) SDI (G3 P1) SDI (G3 P2) SDI (G4 P1) SDI (G4 P2) AES 1 AES 2 Balanced 1 Balanced 2 Unbalanced 1 Unbalanced 2)
input.audio2.frequency	CHOICE		EnumRange (AUTO 48 kHz 32 kHz 24 kHz 16 kHz 12 kHz 8 kHz)
input.audio2.numChannels	CHOICE		EnumRange (LEFT RIGHT DUAL STEREO)
input.audio2.volume	INTEGER		IntegerRange (0 .. 100)
input.audio2.languageCode	CHOICE		EnumRange (None English French Spanish German)

name	*type*	*flags*	*range*
input.audio2.type	CHOICE		EnumRange (None SDI (G1 P1) SDI (G1 P2) SDI (G2 P1) SDI (G2 P2) SDI (G3 P1) SDI (G3 P2) SDI (G4 P1) SDI (G4 P2) AES 1 AES 2 Balanced 1 Balanced 2 Unbalanced 1 Unbalanced 2)
input.audio3.frequency	CHOICE		EnumRange (AUTO 48 kHz 32 kHz 24 kHz 16 kHz 12 kHz 8 kHz)
input.audio3.numChannels	CHOICE		EnumRange (LEFT RIGHT DUAL STEREO)
input.audio3.volume	INTEGER		IntegerRange (0 .. 100)
input.audio3.languageCode	CHOICE		EnumRange (None English French Spanish German)
input.audio3.type	CHOICE		EnumRange (None SDI (G1 P1) SDI (G1 P2) SDI (G2 P1) SDI (G2 P2) SDI (G3 P1) SDI (G3 P2) SDI (G4 P1) SDI (G4 P2) AES 1 AES 2 Balanced 1 Balanced 2 Unbalanced 1 Unbalanced 2)

name	*type*	*flags*	*range*
input.audio4.frequency	CHOICE		EnumRange (AUTO 48 kHz 32 kHz 24 kHz 16 kHz 12 kHz 8 kHz)
input.audio4.numChannels	CHOICE		EnumRange (LEFT RIGHT DUAL STEREO)
input.audio4.volume	INTEGER		IntegerRange (0 .. 100)
input.audio4.languageCode	CHOICE		EnumRange (None English French Spanish German)
input.audio4.type	CHOICE		EnumRange (None SDI (G1 P1) SDI (G1 P2) SDI (G2 P1) SDI (G2 P2) SDI (G3 P1) SDI (G3 P2) SDI (G4 P1) SDI (G4 P2) AES 1 AES 2 Balanced 1 Balanced 2 Unbalanced 1 Unbalanced 2)
output.broadcast.generateStream	CHOICE		EnumRange (OFF ON)
output.broadcast.IPAddress	TEXT		StringRange
output.broadcast.port	INTEGER		IntegerRange (0 .. 65535)
output.broadcast.sdpFileName	TEXT		StringRange
output.broadcast.sessionName	TEXT		StringRange
output.dvb.enabled	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
output.dvb.videoPid	INTEGER		IntegerRange (0 .. 8191)
output.dvb.audioPid	INTEGER		IntegerRange (0 .. 8191)
output.dvb.pmtPid	INTEGER		IntegerRange (0 .. 8191)
output.dvb.pcrPid	INTEGER		IntegerRange (0 .. 8191)
output.dvb.progNum	INTEGER		IntegerRange (0 .. 65535)
output.dvb.burstModeEnabled	CHOICE		EnumRange (SMOOTH BURST)
output.dvb.packetSize204Enabled	CHOICE		EnumRange (188 bytes 204 bytes)
output.dvb.extSyncEnabled	CHOICE		EnumRange (INTERN EXTERN)
output.dvb.tsBitRate	INTEGER		IntegerRange (0 .. 0)
output.dvb.IPAddress	TEXT		StringRange
output.dvb.port	INTEGER		IntegerRange (0 .. 65535)
output.dvb.multicastTTL	INTEGER		IntegerRange (0 .. 0)
output.dvb.tos	INTEGER		IntegerRange (0 .. 255)
output.dvb.interface	CHOICE		EnumRange (AUTO 1 2)
output.dvb.ipEnabled	CHOICE		EnumRange (OFF ON)
advancedOutput.network.networkInterface	CHOICE		EnumRange (AUTO 1 2)

name	*type*	*flags*	*range*
advancedOutput.network.multicastTTL	INTEGER		IntegerRange (0 .. 0)
advancedOutput.network.maxPacketSize	INTEGER		IntegerRange (0 .. 0)
advancedOutput.network.videoToS	INTEGER		IntegerRange (0 .. 255)
advancedOutput.network.audioToS	INTEGER		IntegerRange (0 .. 255)
advancedOutput.network.sapToS	INTEGER		IntegerRange (0 .. 255)
encoding.video.bitrate	INTEGER		IntegerRange (0 .. 0)
encoding.video.maxBitrate	INTEGER		IntegerRange (0 .. 0)
encoding.video.rateControlType	CHOICE		EnumRange (CBR VBR)
encoding.video.codecType	CHOICE		EnumRange (MPEG-4 ASP H264 Main H264 Baseline)
encoding.video.useOverscan	CHOICE		EnumRange (OFF ON)
encoding.audio.bitrate	INTEGER		IntegerRange (0 .. 0)
encoding.audio.codecType	CHOICE		EnumRange (MPEG-4 AAC MPEG-4 HE-AAC MPEG-2 AAC MPEG-2 HE-AAC MPEG-1 LAYER2)
encoding.audio2.bitrate	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
encoding.audio2.codecType	CHOICE		EnumRange (MPEG-4 AAC MPEG-4 HE-AAC MPEG-2 AAC MPEG-2 HE-AAC MPEG-1 LAYER2)
encoding.audio3.bitrate	INTEGER		IntegerRange (0 .. 0)
encoding.audio3.codecType	CHOICE		EnumRange (MPEG-4 AAC MPEG-4 HE-AAC MPEG-2 AAC MPEG-2 HE-AAC MPEG-1 LAYER2)
encoding.audio4.bitrate	INTEGER		IntegerRange (0 .. 0)
encoding.audio4.codecType	CHOICE		EnumRange (MPEG-4 AAC MPEG-4 HE-AAC MPEG-2 AAC MPEG-2 HE-AAC MPEG-1 LAYER2)
advancedEncoding.general.keyFramePeriod	FLOAT		DoubleRange (0.00 .. 0.00)
advancedEncoding.general.BFramePeriod	CHOICE		EnumRange (OFF ON)
advancedEncoding.general.aspectRatio	CHOICE		EnumRange (4:3 16:9 4:3 Letterbox 16:9 Pillarbox)
advancedEncoding.general.closedCaptions	CHOICE		EnumRange (OFF ON)
advancedEncoding.filters.useSpatialFilter	CHOICE		EnumRange (OFF ON)
advancedEncoding.rateControl.videoBufferSize	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
encoding.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
encoding.maxBitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stop	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
status	TEXT	R/O	StringRange R/O
internal.status	TEXT	R/O	StringRange R/O
internal.audioBitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (4CASTER)

9.229 Envivio-4View-D1

This page describes the device driver and the device window for the Envivio 4View D1 MPEG-4 video decoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Decoder settings
- --- Decoder readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The only configuration parameter which must be set for the Envivio decoder is it's IP address:

parameter	*description*
address	Enter the IP address of the decoder here followed by a blanc character and the SNMP community string set at the decoder. Example:

parameter	*description*
10.0.0.1 private	

Remarks

- To change parameters at the device, the decoder must be in stopped mode. The changes will be applied when the decoder is started again.
- Loading and saving presets for this device is only possible if the device is in STOPPED mode.
- The decoder does not provide a true summary fault indication. The driver reads the 'lastFault' parameter from the device and displays this at both device window pages. The '4VIEW' fault flag is set by the driver while the 'lastFault' message is not empty. This leads to a latching behaviour for some faults: If e.g. you mistyped the input URL, the driver states this fault until the decoder properly decodes another signal. This is exactly how the web user interface of the device behaves.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.status	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.description	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mibVersion	TEXT	R/O	StringRange R/O
input.url	TEXT		StringRange

name	*type*	*flags*	*range*
input.test	CHOICE		EnumRange (75% COLOR BARS 60% SWEEP)
video.tvStandard	CHOICE		EnumRange (NTSC PAL)
video.nativeSize	CHOICE		EnumRange (OFF ON)
video.deblockingFilter	CHOICE		EnumRange (OFF ON)
video.wss	CHOICE		EnumRange (OFF ON)
transport.protocol	CHOICE		EnumRange (UDP TCP HTTP)
transport.buffer	FLOAT		DoubleRange (0.00 .. 0.00)
transport.rtspProxy	CHOICE		EnumRange (OFF ON)
transport.rtspProxName	TEXT		StringRange
transport.rtspProxPort	INTEGER		IntegerRange (0 .. 0)
transport.httpProxy	CHOICE		EnumRange (OFF ON)
transport.httpProxName	TEXT		StringRange
transport.httpProxPort	INTEGER		IntegerRange (0 .. 0)
genlock.source	CHOICE		EnumRange (FREERUN EXTERNAL)
genlock.horizOffset	INTEGER		IntegerRange (0 .. 1727)
genlock.vertOffset	INTEGER		IntegerRange (0 .. 625)
output.type	CHOICE		EnumRange (COMPOSITE RGB YUV)
state.url	TEXT	R/O	StringRange R/O
state.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.video.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.video.props	TEXT	R/O	StringRange R/O
state.audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.audio.1.props	TEXT	R/O	StringRange R/O
state.audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.audio.2.props	TEXT	R/O	StringRange R/O
state.audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
state.audio.3.props	TEXT	R/O	StringRange R/O
state.audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.audio.4.props	TEXT	R/O	StringRange R/O
status	TEXT	R/O	StringRange R/O
lastError	TEXT	R/O	StringRange R/O
start	TEXT		StringRange
stop	TEXT		StringRange
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (4VIEW)

9.230 Ericsson-AVP

This page describes the device driver and the device window for the Ericsson AVP Encoder / Satellite Modulator.

The device driver expects the following modules to be installed in the device:

- ASI IO Option Card
- CE-xH42 Pre-Processor
- CE-xH42 Encoder
- Satellite Modulator Option Card

The device driver automatically identifies the installed cards and configures itself accordingly. If the satellite modulator card is missing, all modulator related parameters are disabled, the parameter pages which solely contain modulator parameters are removed.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- Satellite output transport stream parameters
- --- ASI output transport stream parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters

- --- Audio 4 parameters
- --- Audio 5 parameters
- --- Audio 6 parameters
- --- Audio 7 parameters
- --- Audio 8 parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Parameters are grouped / arranged in almost the same way as the Voyagers web GUI does.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder. Pleased note that the modulator related configuration parameters are disabled unless the driver detects the presence of a satellite modulator card.

parameter	***description***
address	The encoder's IP address in 'dotted quad notation', e.g. **10.0.0.1**
frequencyInputMode	Selects if the TX frequency via the L-band output shall be set/shown in terms of L-band frequency or uplink frequency. In the latter case the upconverter frequency parameter below is taken into account.
upConverterFrequency	The upconverter frequency to be added to the L-band output frequency when the mode setting above is 'Uplink Frequency'.
upConverterPower	Switches the BUC power supply via the L-band cable.
upConverterReference	Switches the frequency reference sent to the upconverter via the L-band cable.
clockReferenceMode	Selects the clock reference used by the modulator.

Fault flags

The Ericsson AVP encoder internally knows over 250 different fault conditions. The sat-nms device driver shows a selection of important faults and a fault called "Other fault". The latter is raised if the encoder shows a summary fault but none of the fault flags known to the sat-nms software is set.

Please note, that masking a fault condition at the encoder's web GUI affects the encoder's summary fault but not the fault as it is shown in the software. To mask a fault flag in the sat-nms software, use the device's 'Faults' page and set the fault priority of this fault flag to INFO or OFF.

Status polling

To improve the performance of the device driver, the polling cycles at which configuration parameters are read back have been set to 40-70 seconds. Polling parameters faster would degrade the performance of the encoder's web interface and als slow down the detection of faults.

As a consequence of this, it may take up to 70 seconds until a setting changed at the encoder's web interface becomes visible in the sat-nms software.

Transport stream mux

The sat-nms device driver is not capable to edit all aspects of transport stream parameters. Specially, it is not possible to add or remove services from transport streams.

The device driver assumes that there is one transport stream with one service for each output and lets you edit the parameters of these streams / services. This completely covers the requirements of a SNG use of the device.

For more complex configurations you may use the encoder's web interface. The parameters of the first service are still editable in the sat-nms software. If the first and only service for a transport stream is removed in the encoder's web interface, the sat-nms software shows a communication fault as the parameters of this service are no longer existent. Add a dummy service to the transport stream in the web interface if this happens.

BISS key management

Unlike the Voyager-II encoder, the AVP does not permit to assign BISS keys / session words directly to an output stream. Instead, it uses an internal of up to 31 keys which are assigned to the stream by number.

The sat-nms device driver always uses the BISS key[1] table entry, emulating the old behaviour. This is necessary to make the device work together with the BISS key central database (CDB).

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Com
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.assemblyDate	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.serialNumber	TEXT	R/O	StringRange R/O
info.alarmStatus	TEXT	R/O	StringRange R/O
info.slot.1	TEXT	R/O	StringRange R/O
info.slot.2	TEXT	R/O	StringRange R/O
info.slot.3	TEXT	R/O	StringRange R/O
info.slot.4	TEXT	R/O	StringRange R/O
info.slot.5	TEXT	R/O	StringRange R/O
info.slot.6	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true fa
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 1000
config.frequencyInputMode	CHOICE	SETUP	EnumRange DIS (IF Freque Frequency)
config.upConverterFrequency	FLOAT	SETUP	DoubleRange DIS (5000.0(
config.upConverterPower	CHOICE	SETUP	EnumRange DIS (Off 15V 2
config.upConverterReference	CHOICE	SETUP	EnumRange DIS (Off On)
config.clockReferenceMode	CHOICE	SETUP	EnumRange DIS (Internal E
config.noSlots	CHOICE	SAVE SETUP	EnumRange (4 6)
config.deviceComment	TEXT	SAVE SETUP	StringRange
mod.DVBS2Mode	CHOICE		EnumRange DIS (CCM VCM
mod.FECRate	CHOICE		EnumRange DIS (1/4 1/3 2 5/6 7/8 8/9 9/10)
mod.frameSize	CHOICE		EnumRange DIS (Normal S

name	*type*	*flags*	*range*
mod.IFFrequency	FLOAT		DoubleRange DIS (50.000
mod.IFPower	FLOAT		DoubleRange DIS (-30.0 ..
mod.IFSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.inputBufferSize	INTEGER		IntegerRange DIS (512 .. 4
mod.inputFormat	CHOICE		EnumRange DIS (MPEG TS
mod.inputMode	CHOICE		EnumRange DIS (Manual S Automatic Dual TS)
mod.LBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.LBandSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.modulation	CHOICE		EnumRange DIS (QPSK 8PS 32APSK)
mod.modulationStandard	CHOICE		EnumRange DIS (DVB-S DV
mod.modulationState	CHOICE		EnumRange DIS (Off On)
mod.modulationTest	CHOICE		EnumRange DIS (Normal H Full-rate Reversals Impulse
mod.NCRStampingPID	INTEGER		IntegerRange DIS (0 .. 819
mod.outputPowerUpState	CHOICE		EnumRange DIS (On (Redu State)
mod.outputSelect	CHOICE		EnumRange DIS (IF L-Banc
mod.outputState	CHOICE		EnumRange DIS (Off On (R (Nominal Power))
mod.pilots	CHOICE		EnumRange DIS (Off On)
mod.primaryInputSource	CHOICE		EnumRange DIS (Stream 1
mod.rasEnable	CHOICE		EnumRange DIS (Off On)
mod.rasKey	TEXT		StringRange DIS
mod.rasVersion	TEXT	R/O	StringRange R/O DIS
mod.rateAdaptation	CHOICE		EnumRange DIS (Off On)
mod.reducedIFPower	FLOAT		DoubleRange DIS (-30.0 ..

name	*type*	*flags*	*range*
mod.reducedLBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.rollOffFactor	CHOICE		EnumRange DIS (20 % 25
mod.symbolMappingMode	CHOICE		EnumRange DIS (Peak Pow
mod.symbolRate	FLOAT		DoubleRange DIS (0.1320(
mod.TSInputSelect	CHOICE		EnumRange DIS (Primary S
mod.TXFrequency	FLOAT		DoubleRange DIS (10950.(
stream.mod.enabled	TEXT	R/O	StringRange R/O
stream.mod.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.networkName	TEXT		StringRange DIS
stream.mod.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 100)
stream.mod.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.mod.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.mod.serviceProvider	TEXT		StringRange DIS
stream.mod.serviceName	TEXT		StringRange DIS
stream.mod.serviceID	TEXT		StringRange DIS
stream.mod.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.mod.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.mod.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.scrambling	CHOICE		EnumRange DIS (Off On)
stream.mod.bissSW	TEXT	NOPRESET	StringRange DIS
stream.mod.bissMode	CHOICE		EnumRange DIS (Mode-1 M
stream.asi.enabled	TEXT	R/O	StringRange R/O DIS
stream.asi.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.networkName	TEXT		StringRange DIS

name	*type*	*flags*	*range*
stream.asi.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 400
stream.asi.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.asi.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.asi.serviceProvider	TEXT		StringRange DIS
stream.asi.serviceName	TEXT		StringRange DIS
stream.asi.serviceID	TEXT		StringRange DIS
stream.asi.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.asi.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.asi.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.scrambling	CHOICE		EnumRange DIS (Off On)
stream.asi.bissSW	TEXT	NOPRESET	StringRange DIS
stream.asi.bissMode	CHOICE		EnumRange DIS (Mode-1 M
video.inputLocked	TEXT	R/O	StringRange R/O
video.onVideoLoss	CHOICE		EnumRange (Freeze Black
video.source	CHOICE		EnumRange (SDI Bars Blac
video.standard	CHOICE		EnumRange (576i25 480i2 1080i25 1080i29.97 720p5
video.horizontalBandwidth	CHOICE		EnumRange (30 40 50 60 (passthru))
video.adaptivePreprocessing	CHOICE		EnumRange (Off On)
video.stillsDetection	CHOICE		EnumRange (Off Intelligen
video.inputDetection	CHOICE		EnumRange (Off autoDete autoDetectConfigure)

name	*type*	*flags*	*range*
video.profile	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.profile.sd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE
video.profile.hd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.outputResolution	CHOICE		EnumRange (128 x 96 352 480 480 x 480 528 x 480 5 704 x 480 720 x 480 352 x 480 x 576 528 x 576 544 x 576 720 x 576 960 x 720 1 1080 1920 x 1080 Full)

name	*type*	*flags*	*range*
video.bitrate	FLOAT		DoubleRange (1.500 .. 50.
video.bufferMode	CHOICE		EnumRange (CBR Seamles Low Delay)
video.aspectRatio	CHOICE		EnumRange (4:3 16:9)
video.gopLength	INTEGER		IntegerRange (4 .. 250)
video.bFrames	CHOICE		EnumRange (I IP IBP IBBP
video.sceneCutDetection	CHOICE		EnumRange (Off On)
video.delay	TEXT	R/O	StringRange R/O
video.maxCABACBitRate	FLOAT		DoubleRange (30.000 .. 62
video.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
video.copyright	CHOICE		EnumRange (Off On)
video.original	CHOICE		EnumRange (Off On)
video.pesPacketPer	CHOICE		EnumRange (GOP Picture)
video.auCtrl	CHOICE		EnumRange (Off RAI RAI a Info and TC RAI, AU Info, T
audio.1.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.1.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.1.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.1.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.1.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.1.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.1.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.1.auCtrl	CHOICE		EnumRange (Off AU Info)

name	*type*	*flags*	*range*
audio.1.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.1.channelId	INTEGER		IntegerRange (0 .. 255)
audio.1.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.1.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.1.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.1.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.1.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.1.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.1.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.preset	CHOICE		EnumRange (Subtle Stand
audio.1.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.1.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.1.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.1.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.1.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.1.copyright	CHOICE		EnumRange (Off On)
audio.1.original	CHOICE		EnumRange (Off On)
audio.1.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.1.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.1.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.1.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.1.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.2.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.2.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.2.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.2.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.2.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.2.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.2.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.2.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.2.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.2.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.2.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.2.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.2.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.2.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.2.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.2.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.2.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.preset	CHOICE		EnumRange (Subtle Stand
audio.2.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.2.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.2.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.2.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.2.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.2.copyright	CHOICE		EnumRange (Off On)
audio.2.original	CHOICE		EnumRange (Off On)
audio.2.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.2.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.2.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.2.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.2.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.3.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.3.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.3.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.3.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.3.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.3.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.3.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.3.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.3.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.3.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.3.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.3.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.3.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.3.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.3.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.3.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.3.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.preset	CHOICE		EnumRange (Subtle Stand
audio.3.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.3.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.3.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.3.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.3.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.3.copyright	CHOICE		EnumRange (Off On)
audio.3.original	CHOICE		EnumRange (Off On)
audio.3.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.3.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.3.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.3.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.3.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.4.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.4.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.4.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.4.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.4.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.4.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.4.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.4.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.4.bitDepth	CHOICE		EnumRange (16-bit 20-bit

name	*type*	*flags*	*range*
audio.4.channelId	INTEGER		IntegerRange (0 .. 255)
audio.4.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.4.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.4.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.4.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.4.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.4.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.4.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.preset	CHOICE		EnumRange (Subtle Stand
audio.4.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.4.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.4.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.4.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.4.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.4.copyright	CHOICE		EnumRange (Off On)
audio.4.original	CHOICE		EnumRange (Off On)
audio.4.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.4.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.4.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.4.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.4.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.5.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.5.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.5.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.5.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.5.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.5.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.5.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.5.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.5.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.5.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.5.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.5.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.5.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.5.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.5.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.5.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.5.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.preset	CHOICE		EnumRange (Subtle Stand
audio.5.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.5.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.5.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.5.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.5.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.5.copyright	CHOICE		EnumRange (Off On)
audio.5.original	CHOICE		EnumRange (Off On)
audio.5.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.5.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.5.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.5.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.5.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.6.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.6.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.6.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.6.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.6.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.6.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.6.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.6.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.6.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.6.channelId	INTEGER		IntegerRange (0 .. 255)
audio.6.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.6.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.6.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)

name	*type*	*flags*	*range*
audio.6.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.6.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.6.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.6.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.preset	CHOICE		EnumRange (Subtle Stand
audio.6.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.6.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.6.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.6.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.6.copyright	CHOICE		EnumRange (Off On)
audio.6.original	CHOICE		EnumRange (Off On)
audio.6.deEmphasis	CHOICE		EnumRange (Off 50/15us

name	*type*	*flags*	*range*
audio.6.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.6.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.6.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.6.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.7.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.7.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.7.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.7.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.7.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.7.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.7.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.7.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.7.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.7.channelId	INTEGER		IntegerRange (0 .. 255)
audio.7.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.7.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.7.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.7.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.7.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.7.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.7.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)

name	*type*	*flags*	*range*
audio.7.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.7.preset	CHOICE		EnumRange (Subtle Stand
audio.7.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.7.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.7.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.7.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.7.copyright	CHOICE		EnumRange (Off On)
audio.7.original	CHOICE		EnumRange (Off On)
audio.7.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.7.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.7.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.7.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.7.silencePeriod	INTEGER		IntegerRange (0 .. 300)

name	*type*	*flags*	*range*
audio.8.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.8.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.8.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.8.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.8.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.8.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.8.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.8.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.8.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.8.channelId	INTEGER		IntegerRange (0 .. 255)
audio.8.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.8.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.8.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.8.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.8.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.8.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.8.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.preset	CHOICE		EnumRange (Subtle Stand
audio.8.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.8.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)

name	*type*	*flags*	*range*
audio.8.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.8.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.8.copyright	CHOICE		EnumRange (Off On)
audio.8.original	CHOICE		EnumRange (Off On)
audio.8.deEmphasis	CHOICE		EnumRange (Off 50/15us C
audio.8.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.8.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.8.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.8.silencePeriod	INTEGER		IntegerRange (0 .. 300)
internal.x	FLOAT	R/O	DoubleRange R/O (0.0000
slot.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.m	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.t	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ASI c
faults.02	ALARM	R/O	AlarmFlagRange R/O (ASI c
faults.03	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pre-p
faults.05	ALARM	R/O	AlarmFlagRange R/O (Pre-p
faults.06	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.09	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.10	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.12	ALARM	R/O	AlarmFlagRange R/O (Over
faults.13	ALARM	R/O	AlarmFlagRange R/O (Othe

9.231 Ericsson-Voyager-II-v9.5

This page describes the device driver and the device window for the Ericsson Voyager II Encoder / Satellite Modulator with SW version. This driver supports firmware versions up to **9.5** (at least) in the old hardware chassis. For units with firmware version **9.15** and newer use Ericsson-Voyager-II . For the new hardware platform AVP 2000 Contribution Encoders and AVP 3000 Voyager SNG Encoders use driver Ericsson-AVP (firmware **9.19** an newer, Hardware Version **10.0** and newer}

The device driver expects the following modules to be installed in the device:

- ASI IO Option Card
- CE-xH42 Pre-Processor
- CE-xH42 Encoder
- Satellite Modulator Option Card

The device driver automatically identifies the installed cards and configures itself accordingly. If the satellite modulator card is missing, all modulator related parameters are disabled, the parameter pages which solely contain modulator parameters are removed.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- Satellite output transport stream parameters
- --- ASI output transport stream parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- Audio 5 parameters
- --- Audio 6 parameters
- --- Audio 7 parameters
- --- Audio 8 parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Parameters are grouped / arranged in almost the same way as the Voyagers web GUI does.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder. Pleased note that the modulator related configuration parameters are disabled unless the driver detects the presence of a satellite modulator card.

parameter	***description***
address	The encoder's IP address in 'dotted quad notation', e.g. **10.0.0.1**
frequencyInputMode	Selects if the TX frequency via the L-band output shall be set/shown in terms of L-band frequency or uplink frequency. In the latter case the upconverter frequency parameter below is taken into account.
upConverterFrequency	The upconverter frequency to be added to the L-band output frequency when the mode setting above is 'Uplink Frequency'.
upConverterPower	Switches the BUC power supply via the L-band cable.
upConverterReference	Switches the frequency reference sent to the upconverter via the L-band cable.

parameter	*description*
clockReferenceMode	Selects the clock reference used by the modulator.

Fault flags

The Voyager-II encoder internally knows over 250 different fault conditions. The sat-nms device driver shows a selection of important faults and a fault called "Other fault". The latter is raised if the encoder shows a summary fault but none of the fault flags known to the sat-nms software is set.

Please note, that masking a fault condition at the encoder's web GUI affects the encoder's summary fault but not the fault as it is shown in the software. To mask a fault flag in the sat-nms software, use the device's 'Faults' page and set the fault priority of this fault flag to INFO or OFF.

Status polling

To improve the performance of the device driver, the polling cycles at which configuration parameters are read back have been set to 40-70 seconds. Polling parameters faster would degrade the performance of the encoder's web interface and als slow down the detection of faults.

As a consequence of this, it may take up to 70 seconds until a setting changed at the encoder's web interface becomes visible in the sat-nms software.

Transport stream mux

The sat-nms device driver is not capable to edit all aspects of transport stream parameters. Specially, it is not possible to add or remove services from transport streams.

The device driver assumes that there is one transport stream with one service for each output and lets you edit the parameters of these streams / services. This completely covers the requirements of a SNG use of the device.

For more complex configurations you may use the encoder's web interface. The parameters of the first service are still editable in the sat-nms software. If the first and only service for a transport stream is removed in the encoder's web interface, the sat-nms software shows a communication fault as the parameters of this service are no longer existent. Add a dummy service to the transport stream in the web interface if this happens.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Com
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.assemblyDate	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.serialNumber	TEXT	R/O	StringRange R/O
info.alarmStatus	TEXT	R/O	StringRange R/O
info.slot.1	TEXT	R/O	StringRange R/O
info.slot.2	TEXT	R/O	StringRange R/O
info.slot.3	TEXT	R/O	StringRange R/O
info.slot.4	TEXT	R/O	StringRange R/O
info.slot.5	TEXT	R/O	StringRange R/O
info.slot.6	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true fa
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 1000
config.frequencyInputMode	CHOICE	SETUP	EnumRange DIS (IF Freque Frequency)
config.upConverterFrequency	FLOAT	SETUP	DoubleRange DIS (5000.0(
config.upConverterPower	CHOICE	SETUP	EnumRange DIS (Off 15V 2
config.upConverterReference	CHOICE	SETUP	EnumRange DIS (Off On)
config.clockReferenceMode	CHOICE	SETUP	EnumRange DIS (Internal E
config.deviceComment	TEXT	SAVE SETUP	StringRange
mod.DVBS2Mode	CHOICE		EnumRange DIS (CCM VCM

name	*type*	*flags*	*range*
mod.FECRate	CHOICE		EnumRange DIS (1/4 1/3 2 5/6 7/8 8/9 9/10)
mod.frameSize	CHOICE		EnumRange DIS (Normal S
mod.IFFrequency	FLOAT		DoubleRange DIS (50.000
mod.IFPower	FLOAT		DoubleRange DIS (-30.0 ..
mod.IFSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.inputBufferSize	INTEGER		IntegerRange DIS (512 .. 4
mod.inputFormat	CHOICE		EnumRange DIS (MPEG TS
mod.inputMode	CHOICE		EnumRange DIS (Manual S Automatic Dual TS)
mod.LBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.LBandSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.modulation	CHOICE		EnumRange DIS (QPSK 8PS 32APSK)
mod.modulationStandard	CHOICE		EnumRange DIS (DVB-S DV
mod.modulationState	CHOICE		EnumRange DIS (Off On)
mod.modulationTest	CHOICE		EnumRange DIS (Normal H Full-rate Reversals Impulse
mod.NCRStampingPID	INTEGER		IntegerRange DIS (0 .. 819
mod.outputPowerUpState	CHOICE		EnumRange DIS (On (Redu State)
mod.outputSelect	CHOICE		EnumRange DIS (IF L-Banc
mod.outputState	CHOICE		EnumRange DIS (Off On (R (Nominal Power))
mod.pilots	CHOICE		EnumRange DIS (Off On)
mod.primaryInputSource	CHOICE		EnumRange DIS (ASI1 ASI2 Null Packets PRBS)
mod.rasEnable	CHOICE		EnumRange DIS (Off On)
mod.rasKey	TEXT		StringRange DIS

name	*type*	*flags*	*range*
mod.rasVersion	TEXT	R/O	StringRange R/O DIS
mod.rateAdaptation	CHOICE		EnumRange DIS (Off On)
mod.reducedIFPower	FLOAT		DoubleRange DIS (-30.0 ..
mod.reducedLBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.rollOffFactor	CHOICE		EnumRange DIS (20 % 25
mod.secondaryInputSource	CHOICE		EnumRange DIS (ASI1 ASI2 Null Packets PRBS)
mod.symbolMappingMode	CHOICE		EnumRange DIS (Peak Pow
mod.symbolRate	FLOAT		DoubleRange DIS (0.1320(
mod.TSInputSelect	CHOICE		EnumRange DIS (Primary S
mod.TXFrequency	FLOAT		DoubleRange DIS (10950.(
stream.mod.enabled	TEXT	R/O	StringRange R/O
stream.mod.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.networkName	TEXT		StringRange DIS
stream.mod.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 100)
stream.mod.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.mod.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.mod.serviceProvider	TEXT		StringRange DIS
stream.mod.serviceName	TEXT		StringRange DIS
stream.mod.serviceID	TEXT		StringRange DIS
stream.mod.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.mod.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.mod.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.scrambling	CHOICE		EnumRange DIS (Off BISS

name	*type*	*flags*	*range*
stream.mod.bissSW	TEXT	NOPRESET	StringRange DIS
stream.mod.bissESW	TEXT	NOPRESET	StringRange DIS
stream.asi.enabled	TEXT	R/O	StringRange R/O DIS
stream.asi.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.networkName	TEXT		StringRange DIS
stream.asi.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.mode	CHOICE		EnumRange DIS (CBR VBR
stream.asi.siLevel	CHOICE		EnumRange DIS (Off On (P (PAT/PMT/TSDT) On (PAT/P (PAT/PMT/SDT/CAT/NIT) Or (PAT/PMT/TSDT/SDT/CAT/N
stream.asi.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 400
stream.asi.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.asi.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.asi.serviceProvider	TEXT		StringRange DIS
stream.asi.serviceName	TEXT		StringRange DIS
stream.asi.serviceID	TEXT		StringRange DIS
stream.asi.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.asi.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.asi.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.scrambling	CHOICE		EnumRange DIS (Off BISS
stream.asi.bissSW	TEXT	NOPRESET	StringRange DIS
stream.asi.bissESW	TEXT	NOPRESET	StringRange DIS
video.inputLocked	TEXT	R/O	StringRange R/O
video.onVideoLoss	CHOICE		EnumRange (Freeze Black
video.source	CHOICE		EnumRange (SDI Bars Blac

name	*type*	*flags*	*range*
video.standard	CHOICE		EnumRange (576i25 480i2 1080i25 1080i29.97 720p5
video.horizontalBandwidth	CHOICE		EnumRange (30 40 50 60 (passthru))
video.adaptivePreprocessing	CHOICE		EnumRange (Off On)
video.stillsDetection	CHOICE		EnumRange (Off Intelligen
video.inputDetection	CHOICE		EnumRange (Off autoDete autoDetectConfigure)
video.profile	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.profile.sd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE

name	*type*	*flags*	*range*
video.profile.hd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.outputResolution	CHOICE		EnumRange (128 x 96 352 480 480 x 480 528 x 480 5 704 x 480 720 x 480 352 x 480 x 576 528 x 576 544 x 576 720 x 576 960 x 720 1 1080 1920 x 1080 Full)
video.bitrate	FLOAT		DoubleRange (1.500 .. 50.
video.bufferMode	CHOICE		EnumRange (CBR Seamles Low Delay)
video.aspectRatio	CHOICE		EnumRange (4:3 16:9)
video.gopLength	INTEGER		IntegerRange (4 .. 250)
video.bFrames	CHOICE		EnumRange (I IP IBP IBBP I
video.sceneCutDetection	CHOICE		EnumRange (Off On)
video.delay	TEXT	R/O	StringRange R/O
video.maxCABACBitRate	FLOAT		DoubleRange (30.000 .. 62
video.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
video.copyright	CHOICE		EnumRange (Off On)
video.original	CHOICE		EnumRange (Off On)
video.pesPacketPer	CHOICE		EnumRange (GOP Picture)

name	*type*	*flags*	*range*
video.auCtrl	CHOICE		EnumRange (Off RAI RAI a Info and TC RAI, AU Info, T
audio.1.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.1.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.1.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.1.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.1.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.1.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.1.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.1.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.1.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.1.channelId	INTEGER		IntegerRange (0 .. 255)
audio.1.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.1.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.1.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.1.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.1.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.1.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.1.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.preset	CHOICE		EnumRange (Subtle Stand
audio.1.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)

name	*type*	*flags*	*range*
audio.1.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.1.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.1.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.1.copyright	CHOICE		EnumRange (Off On)
audio.1.original	CHOICE		EnumRange (Off On)
audio.1.deEmphasis	CHOICE		EnumRange (Off 50/15us (
audio.1.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.1.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.1.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.1.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.2.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)

name	*type*	*flags*	*range*
audio.2.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.2.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.2.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.2.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.2.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.2.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.2.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.2.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.2.channelId	INTEGER		IntegerRange (0 .. 255)
audio.2.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.2.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.2.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.2.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.2.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.2.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.2.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.preset	CHOICE		EnumRange (Subtle Stand
audio.2.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.2.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.2.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.2.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE(
audio.2.copyright	CHOICE		EnumRange (Off On)
audio.2.original	CHOICE		EnumRange (Off On)
audio.2.deEmphasis	CHOICE		EnumRange (Off 50/15us (
audio.2.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.2.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.2.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.2.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.3.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.3.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.3.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.3.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.3.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.3.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.3.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.3.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.3.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.3.channelId	INTEGER		IntegerRange (0 .. 255)
audio.3.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.3.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.3.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.3.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.3.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.3.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.3.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.preset	CHOICE		EnumRange (Subtle Stand
audio.3.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.3.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.3.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.3.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE(
audio.3.copyright	CHOICE		EnumRange (Off On)
audio.3.original	CHOICE		EnumRange (Off On)
audio.3.deEmphasis	CHOICE		EnumRange (Off 50/15us (
audio.3.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.3.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.3.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.3.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.4.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.4.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.4.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.4.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.4.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.4.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.4.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.4.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.4.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.4.channelId	INTEGER		IntegerRange (0 .. 255)
audio.4.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.4.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.4.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.4.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.4.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.4.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.4.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.preset	CHOICE		EnumRange (Subtle Stand
audio.4.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.4.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.4.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.4.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.4.copyright	CHOICE		EnumRange (Off On)
audio.4.original	CHOICE		EnumRange (Off On)
audio.4.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.4.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.4.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.4.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.4.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.5.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.5.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.5.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.5.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.5.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.5.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.5.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.5.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.5.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.5.channelId	INTEGER		IntegerRange (0 .. 255)
audio.5.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.5.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.5.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.5.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.5.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.5.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.5.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.preset	CHOICE		EnumRange (Subtle Stand
audio.5.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.5.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.5.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.5.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.5.copyright	CHOICE		EnumRange (Off On)
audio.5.original	CHOICE		EnumRange (Off On)
audio.5.deEmphasis	CHOICE		EnumRange (Off 50/15us C
audio.5.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.5.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.5.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.5.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.6.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.6.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.6.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.6.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.6.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.6.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.6.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.6.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.6.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.6.channelId	INTEGER		IntegerRange (0 .. 255)
audio.6.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.6.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.6.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.6.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.6.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.6.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.6.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.preset	CHOICE		EnumRange (Subtle Stand
audio.6.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.6.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.6.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.6.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE(
audio.6.copyright	CHOICE		EnumRange (Off On)
audio.6.original	CHOICE		EnumRange (Off On)
audio.6.deEmphasis	CHOICE		EnumRange (Off 50/15us (
audio.6.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.6.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.6.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.6.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.7.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.7.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.7.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.7.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.7.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.7.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.7.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.7.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.7.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.7.channelId	INTEGER		IntegerRange (0 .. 255)
audio.7.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.7.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.7.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.7.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.7.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.7.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.7.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.7.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.7.preset	CHOICE		EnumRange (Subtle Stand
audio.7.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.7.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.7.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.7.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.7.copyright	CHOICE		EnumRange (Off On)
audio.7.original	CHOICE		EnumRange (Off On)
audio.7.deEmphasis	CHOICE		EnumRange (Off 50/15us C
audio.7.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.7.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.7.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.7.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.8.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.8.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)

name	*type*	*flags*	*range*
audio.8.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.8.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.8.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.8.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.8.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.8.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.8.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.8.channelId	INTEGER		IntegerRange (0 .. 255)
audio.8.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.8.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.8.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.8.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.8.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.8.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.8.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.preset	CHOICE		EnumRange (Subtle Stand
audio.8.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.8.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.8.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.8.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.8.copyright	CHOICE		EnumRange (Off On)
audio.8.original	CHOICE		EnumRange (Off On)
audio.8.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.8.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.8.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.8.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.8.silencePeriod	INTEGER		IntegerRange (0 .. 300)
internal.x	FLOAT	R/O	DoubleRange R/O (0.0000
slot.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.m	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (ASI
faults.02	ALARM	R/O	AlarmFlagRange R/O (ASI
faults.03	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pre-
faults.05	ALARM	R/O	AlarmFlagRange R/O (Pre-
faults.06	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.09	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.10	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ove
faults.13	ALARM	R/O	AlarmFlagRange R/O (Othe

9.232 Ericsson-Voyager-II

This page describes the device driver and the device window for the Ericsson Voyager II Encoder / Satellite Modulator. This driver supports firmware versions up to **9.15** (at least) in the new hardware chassis. For units with firmware version **9.05** and earlier use Ericsson-Voyager-II-v9.5 . For the new hardware platform AVP 2000 Contribution Encoders and AVP 3000 Voyager SNG Encoders use driver Ericsson-AVP (firmware **9.19** an newer, Hardware Version **10.0** and newer}

The device driver expects the following modules to be installed in the device:

- ASI IO Option Card
- CE-xH42 Pre-Processor
- CE-xH42 Encoder
- Satellite Modulator Option Card

The device driver automatically identifies the installed cards and configures itself accordingly. If the satellite modulator card is missing, all modulator related parameters are disabled, the parameter pages which solely contain modulator parameters are removed.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- Satellite output transport stream parameters
- --- ASI output transport stream parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- Audio 5 parameters
- --- Audio 6 parameters
- --- Audio 7 parameters
- --- Audio 8 parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Parameters are grouped / arranged in almost the same way as the Voyagers web GUI does.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder. Pleased note that the modulator related configuration parameters are disabled unless the driver detects the presence of a satellite modulator card.

parameter	***description***
address	The encoder's IP address in 'dotted quad notation', e.g. **10.0.0.1**
frequencyInputMode	Selects if the TX frequency via the L-band output shall be set/shown in terms of L-band frequency or uplink frequency. In the latter case the upconverter frequency parameter below is taken into account.
upConverterFrequency	The upconverter frequency to be added to the L-band output frequency when the mode setting above is 'Uplink Frequency'.
upConverterPower	Switches the BUC power supply via the L-band cable.
upConverterReference	Switches the frequency reference sent to the upconverter via the L-band cable.
clockReferenceMode	Selects the clock reference used by the modulator.

Fault flags

The Voyager-II encoder internally knows over 250 different fault conditions. The sat-nms device driver shows a selection of important faults and a fault called "Other fault". The latter is raised if the encoder shows a summary fault but none of the fault flags known to the sat-nms software is set.

Please note, that masking a fault condition at the encoder's web GUI affects the encoder's summary fault but not the fault as it is shown in the software. To mask a fault flag in the sat-nms software, use the device's 'Faults' page and set the fault priority of this fault flag to INFO or OFF.

Status polling

To improve the performance of the device driver, the polling cycles at which configuration parameters are read back have been set to 40-70 seconds. Polling parameters faster would degrade the performance of the encoder's web interface and als slow down the detection of faults.

As a consequence of this, it may take up to 70 seconds until a setting changed at the encoder's web interface becomes visible in the sat-nms software.

Transport stream mux

The sat-nms device driver is not capable to edit all aspects of transport stream parameters. Specially, it is not possible to add or remove services from transport streams.

The device driver assumes that there is one transport stream with one service for each output and lets you edit the parameters of these streams / services. This completely covers the requirements of a SNG use of the device.

For more complex configurations you may use the encoder's web interface. The parameters of the first service are still editable in the sat-nms software. If the first and only service for a transport stream is removed in the encoder's web interface, the sat-nms software shows a communication fault as the parameters of this service are no longer existent. Add a dummy service to the transport stream in the web interface if this happens.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Com
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.assemblyDate	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.serialNumber	TEXT	R/O	StringRange R/O
info.alarmStatus	TEXT	R/O	StringRange R/O
info.slot.1	TEXT	R/O	StringRange R/O
info.slot.2	TEXT	R/O	StringRange R/O
info.slot.3	TEXT	R/O	StringRange R/O
info.slot.4	TEXT	R/O	StringRange R/O
info.slot.5	TEXT	R/O	StringRange R/O
info.slot.6	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true fa
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 1000
config.frequencyInputMode	CHOICE	SETUP	EnumRange DIS (IF Freque Frequency)
config.upConverterFrequency	FLOAT	SETUP	DoubleRange DIS (5000.0(
config.upConverterPower	CHOICE	SETUP	EnumRange DIS (Off 15V 2
config.upConverterReference	CHOICE	SETUP	EnumRange DIS (Off On)
config.clockReferenceMode	CHOICE	SETUP	EnumRange DIS (Internal E
config.deviceComment	TEXT	SAVE SETUP	StringRange
mod.DVBS2Mode	CHOICE		EnumRange DIS (CCM VCM
mod.FECRate	CHOICE		EnumRange DIS (1/4 1/3 2 5/6 7/8 8/9 9/10)
mod.frameSize	CHOICE		EnumRange DIS (Normal S

name	*type*	*flags*	*range*
mod.IFFrequency	FLOAT		DoubleRange DIS (50.000
mod.IFPower	FLOAT		DoubleRange DIS (-30.0 ..
mod.IFSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.inputBufferSize	INTEGER		IntegerRange DIS (512 .. 4
mod.inputFormat	CHOICE		EnumRange DIS (MPEG TS
mod.inputMode	CHOICE		EnumRange DIS (Manual S Automatic Dual TS)
mod.LBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.LBandSpectrumSense	CHOICE		EnumRange DIS (Normal I
mod.modulation	CHOICE		EnumRange DIS (QPSK 8PS 32APSK)
mod.modulationStandard	CHOICE		EnumRange DIS (DVB-S DV
mod.modulationState	CHOICE		EnumRange DIS (Off On)
mod.modulationTest	CHOICE		EnumRange DIS (Normal H Full-rate Reversals Impulse
mod.NCRStampingPID	INTEGER		IntegerRange DIS (0 .. 819
mod.outputPowerUpState	CHOICE		EnumRange DIS (On (Redu State)
mod.outputSelect	CHOICE		EnumRange DIS (IF L-Banc
mod.outputState	CHOICE		EnumRange DIS (Off On (R (Nominal Power))
mod.pilots	CHOICE		EnumRange DIS (Off On)
mod.primaryInputSource	CHOICE		EnumRange DIS (Stream 1
mod.rasEnable	CHOICE		EnumRange DIS (Off On)
mod.rasKey	TEXT		StringRange DIS
mod.rasVersion	TEXT	R/O	StringRange R/O DIS
mod.rateAdaptation	CHOICE		EnumRange DIS (Off On)
mod.reducedIFPower	FLOAT		DoubleRange DIS (-30.0 ..

name	*type*	*flags*	*range*
mod.reducedLBandPower	FLOAT		DoubleRange DIS (-40.0 ..
mod.rollOffFactor	CHOICE		EnumRange DIS (20 % 25
mod.symbolMappingMode	CHOICE		EnumRange DIS (Peak Pow
mod.symbolRate	FLOAT		DoubleRange DIS (0.1320(
mod.TSInputSelect	CHOICE		EnumRange DIS (Primary S
mod.TXFrequency	FLOAT		DoubleRange DIS (10950.(
stream.mod.enabled	TEXT	R/O	StringRange R/O
stream.mod.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.networkName	TEXT		StringRange DIS
stream.mod.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.mod.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 400
stream.mod.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.mod.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.mod.serviceProvider	TEXT		StringRange DIS
stream.mod.serviceName	TEXT		StringRange DIS
stream.mod.serviceID	TEXT		StringRange DIS
stream.mod.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.mod.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.mod.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.mod.scrambling	CHOICE		EnumRange DIS (Off On)
stream.mod.bissSW	TEXT	NOPRESET	StringRange DIS
stream.mod.bissMode	CHOICE		EnumRange DIS (Mode-1 M
stream.asi.enabled	TEXT	R/O	StringRange R/O DIS
stream.asi.transportStreamID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.networkName	TEXT		StringRange DIS

name	*type*	*flags*	*range*
stream.asi.networkID	INTEGER		IntegerRange DIS (1 .. 655
stream.asi.pmtRepetitionPeriod	CHOICE		EnumRange DIS (500 400
stream.asi.bitrate	FLOAT		DoubleRange DIS (0.0400(
stream.asi.opRateTrackingMode	CHOICE		EnumRange DIS (Off maxV
stream.asi.serviceProvider	TEXT		StringRange DIS
stream.asi.serviceName	TEXT		StringRange DIS
stream.asi.serviceID	TEXT		StringRange DIS
stream.asi.serviceType	CHOICE		EnumRange DIS (Digital TV Teletext Mosaic Data Broa
stream.asi.pmtPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.logicalChannel	INTEGER		IntegerRange DIS (0 .. 0)
stream.asi.pcrPid	INTEGER		IntegerRange DIS (32 .. 81
stream.asi.scrambling	CHOICE		EnumRange DIS (Off On)
stream.asi.bissSW	TEXT	NOPRESET	StringRange DIS
stream.asi.bissMode	CHOICE		EnumRange DIS (Mode-1 M
video.inputLocked	TEXT	R/O	StringRange R/O
video.onVideoLoss	CHOICE		EnumRange (Freeze Black Freeze + Ident Black + Ide + Ident)
video.source	CHOICE		EnumRange (SDI Backplan Black Moving Object Slate)
video.standard	CHOICE		EnumRange (576i25 480i2 1080p59.94 1080i25 1080 720p59.94)
video.horizontalBandwidth	CHOICE		EnumRange (30 40 50 60 (passthru))
video.adaptivePreprocessing	CHOICE		EnumRange (Off On)
video.stillsDetection	CHOICE		EnumRange (Off Intelligen
video.inputDetection	CHOICE		EnumRange (Off autoDete autoDetectConfigure)

name	*type*	*flags*	*range*
video.profile	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.profile.sd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE
video.profile.hd	CHOICE		EnumRange (Off SD_MPEG SD_MPEG2_MP@ML_422_8 SD_H264_MAIN_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HIGH_PROFILE_L SD_H264_HI422P_PROFILE SD_H264_HI422P_PROFILE HD_MPEG2_HP@HL_420_8 HD_MPEG2_HP@HL_422_8 HD_H264_MAIN_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HIGH_PROFILE_L HD_H264_HI422P_PROFILE HD_H264_HI422P_PROFILE
video.outputResolution	CHOICE		EnumRange (128 x 96 352 480 480 x 480 528 x 480 5 704 x 480 720 x 480 352 x 480 x 576 528 x 576 544 x 576 720 x 576 960 x 720 1 1080 1920 x 1080 Full)

name	*type*	*flags*	*range*
video.bitrate	FLOAT		DoubleRange (1.500 .. 50.
video.bufferMode	CHOICE		EnumRange (CBR Seamles Low Delay)
video.aspectRatio	CHOICE		EnumRange (4:3 16:9)
video.gopLength	INTEGER		IntegerRange (4 .. 250)
video.bFrames	CHOICE		EnumRange (I IP IBP IBBP
video.sceneCutDetection	CHOICE		EnumRange (Off On)
video.delay	TEXT	R/O	StringRange R/O
video.maxCABACBitRate	FLOAT		DoubleRange (30.000 .. 62
video.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
video.copyright	CHOICE		EnumRange (Off On)
video.original	CHOICE		EnumRange (Off On)
video.pesPacketPer	CHOICE		EnumRange (GOP Picture)
video.auCtrl	CHOICE		EnumRange (Off RAI RAI a
audio.1.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.1.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.1.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.1.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.1.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.1.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.1.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.1.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.1.bitDepth	CHOICE		EnumRange (16-bit 20-bit

name	*type*	*flags*	*range*
audio.1.channelId	INTEGER		IntegerRange (0 .. 255)
audio.1.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.1.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.1.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.1.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.1.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.1.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.1.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.1.preset	CHOICE		EnumRange (Subtle Stand
audio.1.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.1.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.1.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.1.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.1.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.1.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.1.copyright	CHOICE		EnumRange (Off On)
audio.1.original	CHOICE		EnumRange (Off On)
audio.1.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.1.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.1.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.1.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.1.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.2.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.2.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.2.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.2.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.2.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.2.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.2.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.2.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.2.bitDepth	CHOICE		EnumRange (16-bit 20-bit

name	*type*	*flags*	*range*
audio.2.channelId	INTEGER		IntegerRange (0 .. 255)
audio.2.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.2.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.2.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.2.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.2.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.2.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.2.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.2.preset	CHOICE		EnumRange (Subtle Stand
audio.2.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.2.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.2.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.2.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.2.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.2.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.2.copyright	CHOICE		EnumRange (Off On)
audio.2.original	CHOICE		EnumRange (Off On)
audio.2.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.2.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.2.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.2.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.2.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.3.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.3.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.3.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.3.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.3.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.3.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.3.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.3.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.3.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.3.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.3.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.3.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.3.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.3.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.3.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.3.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.3.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.3.preset	CHOICE		EnumRange (Subtle Stand
audio.3.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.3.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.3.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.3.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.3.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.3.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.3.copyright	CHOICE		EnumRange (Off On)
audio.3.original	CHOICE		EnumRange (Off On)
audio.3.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.3.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.3.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.3.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.3.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.4.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.4.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.4.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.4.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.4.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.4.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.4.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.4.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.4.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.4.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.4.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.4.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.4.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.4.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.4.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.4.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.4.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.4.preset	CHOICE		EnumRange (Subtle Stand
audio.4.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.4.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.4.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.4.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.4.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.4.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.4.copyright	CHOICE		EnumRange (Off On)
audio.4.original	CHOICE		EnumRange (Off On)
audio.4.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.4.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.4.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.4.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.4.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.5.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.5.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.5.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.5.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.5.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.5.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.5.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.5.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.5.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.5.channelId	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
audio.5.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.5.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.5.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.5.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.5.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.5.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.5.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.5.preset	CHOICE		EnumRange (Subtle Stand
audio.5.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.5.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.5.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.5.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.5.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw

name	*type*	*flags*	*range*
audio.5.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.5.copyright	CHOICE		EnumRange (Off On)
audio.5.original	CHOICE		EnumRange (Off On)
audio.5.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.5.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.5.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.5.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.5.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.6.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.6.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.6.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.6.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.6.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.6.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.6.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.6.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.6.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.6.channelId	INTEGER		IntegerRange (0 .. 255)
audio.6.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.6.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.6.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)

name	*type*	*flags*	*range*
audio.6.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.6.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.6.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.6.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.6.preset	CHOICE		EnumRange (Subtle Stand
audio.6.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.6.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.6.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.6.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.6.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.6.copyright	CHOICE		EnumRange (Off On)
audio.6.original	CHOICE		EnumRange (Off On)

name	*type*	*flags*	*range*
audio.6.deEmphasis	CHOICE		EnumRange (Off 50/15us C
audio.6.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.6.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.6.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.6.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.7.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.7.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.7.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.7.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.7.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.7.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.7.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.7.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.7.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.7.channelId	INTEGER		IntegerRange (0 .. 255)
audio.7.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.7.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.7.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.7.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.7.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.7.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)

name	*type*	*flags*	*range*
audio.7.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.7.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.7.preset	CHOICE		EnumRange (Subtle Stand
audio.7.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)
audio.7.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.7.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.7.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.7.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.7.copyright	CHOICE		EnumRange (Off On)
audio.7.original	CHOICE		EnumRange (Off On)
audio.7.deEmphasis	CHOICE		EnumRange (Off 50/15us
audio.7.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.7.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.7.silenceThres	INTEGER		IntegerRange (-90 .. -1)

name	*type*	*flags*	*range*
audio.7.silencePeriod	INTEGER		IntegerRange (0 .. 300)
audio.8.source	CHOICE		EnumRange (Off Mute TT1 SDI1 E4 SDI1 E5 SDI1 E6 S Input 1 Input 2 Input 3 Inp Analogue 2)
audio.8.format	CHOICE		EnumRange (LPCM DolbyD DolbyDigitalPlus DolbyE)
audio.8.codingStandard	CHOICE		EnumRange (MPL2 DD DD ADS)
audio.8.bitrate	CHOICE		EnumRange (16 24 32 48 160 192 224 256 320 384
audio.8.codingMode	CHOICE		EnumRange (10L 10R 10L PA6 PA8)
audio.8.metadataPreset	CHOICE		EnumRange (1 2 3 4 5 6 7
audio.8.embeddedPCR	CHOICE		EnumRange (Off On On (Pi
audio.8.auCtrl	CHOICE		EnumRange (Off AU Info)
audio.8.bitDepth	CHOICE		EnumRange (16-bit 20-bit
audio.8.channelId	INTEGER		IntegerRange (0 .. 255)
audio.8.encapsulation	CHOICE		EnumRange (MPEG2 ADTS LOAS)
audio.8.sbrSignalling	CHOICE		EnumRange (Explicit Impli
audio.8.operatingMode	CHOICE		EnumRange (Off Measure Measure + ALC + LTLC)
audio.8.targetLevel	FLOAT		DoubleRange (-40.0 .. 0.0)
audio.8.longTermPeriod	INTEGER		IntegerRange (5 .. 120)
audio.8.maximumPeak	FLOAT		DoubleRange (-20.0 .. 0.0)
audio.8.maximumGain	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.maximumAttenuation	FLOAT		DoubleRange (0.0 .. 10.0)
audio.8.preset	CHOICE		EnumRange (Subtle Stand
audio.8.resetStats	CHOICE	NOCOMPARE	EnumRange (false true)

name	*type*	*flags*	*range*
audio.8.lipSyncOffset	INTEGER		IntegerRange (-500 .. 500)
audio.8.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.8.language	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageLeft	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.languageRight	CHOICE		EnumRange (afr alb ara ba bul bur cat chi hrv cze dan gae deu gre guj hau hin hu jpn jav kan kor lav lit mac nde nor ori per pan pol por som swa sve tgl tam tel th vie cym xho yor zul one tw
audio.8.mpegVersion	CHOICE		EnumRange (MPEG-1 MPE
audio.8.copyright	CHOICE		EnumRange (Off On)
audio.8.original	CHOICE		EnumRange (Off On)
audio.8.deEmphasis	CHOICE		EnumRange (Off 50/15us (
audio.8.actionCtrlTrackLoss	CHOICE		EnumRange (NoAct BitSv)
audio.8.actionSilence	CHOICE		EnumRange (NoAct BitSv)
audio.8.silenceThres	INTEGER		IntegerRange (-90 .. -1)
audio.8.silencePeriod	INTEGER		IntegerRange (0 .. 300)
internal.x	FLOAT	R/O	DoubleRange R/O (0.0000
slot.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.m	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.t	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ASI c
faults.02	ALARM	R/O	AlarmFlagRange R/O (ASI c
faults.03	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pre-p
faults.05	ALARM	R/O	AlarmFlagRange R/O (Pre-p
faults.06	ALARM	R/O	AlarmFlagRange R/O (CE-x
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.09	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.10	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod
faults.12	ALARM	R/O	AlarmFlagRange R/O (Over
faults.13	ALARM	R/O	AlarmFlagRange R/O (Othe

9.233 Evertz-7700-Frame

This page describes the device driver and the device window for the Evertz 7700 Main Frame with controller via SNMP.

The installed cards are using separated device drivers and should be installed in the same thread in the vlc setup. The main frame driver gives an overview about the installed cards and the frames power supplies status.

The driver was tested with the following SW Releases:

- Kernel: 1.5 2009-09-03
- SW: 3.07 Build 2 2011-01-25

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Overview of installed cards Slot 2-15
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.kernel	TEXT	R/O	StringRange R/O
info.kernelDate	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.versionDate	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
slot.01.card	TEXT	R/O	StringRange R/O
slot.02.card	TEXT	R/O	StringRange R/O
slot.03.card	TEXT	R/O	StringRange R/O
slot.04.card	TEXT	R/O	StringRange R/O
slot.05.card	TEXT	R/O	StringRange R/O
slot.06.card	TEXT	R/O	StringRange R/O
slot.07.card	TEXT	R/O	StringRange R/O
slot.08.card	TEXT	R/O	StringRange R/O
slot.09.card	TEXT	R/O	StringRange R/O
slot.10.card	TEXT	R/O	StringRange R/O
slot.11.card	TEXT	R/O	StringRange R/O
slot.12.card	TEXT	R/O	StringRange R/O
slot.13.card	TEXT	R/O	StringRange R/O
slot.14.card	TEXT	R/O	StringRange R/O
slot.15.card	TEXT	R/O	StringRange R/O
slot.01.name	TEXT	SAVE	StringRange
slot.02.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
slot.03.name	TEXT	SAVE	StringRange
slot.04.name	TEXT	SAVE	StringRange
slot.05.name	TEXT	SAVE	StringRange
slot.06.name	TEXT	SAVE	StringRange
slot.07.name	TEXT	SAVE	StringRange
slot.08.name	TEXT	SAVE	StringRange
slot.09.name	TEXT	SAVE	StringRange
slot.10.name	TEXT	SAVE	StringRange
slot.11.name	TEXT	SAVE	StringRange
slot.12.name	TEXT	SAVE	StringRange
slot.13.name	TEXT	SAVE	StringRange
slot.14.name	TEXT	SAVE	StringRange
slot.15.name	TEXT	SAVE	StringRange
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Frame Status Line)
internal.oid	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.234 Evertz-7703-BPX

This page describes the device driver and the device window for the Evertz 7800 Chassis BPX type redundancy cards including the following supported cards:

- 7703BPX-LB,
- 7703BPXA-LB,
- 7703BPX50-LB,
- 7703BPXA-LB-50,
- 7703BPX-IF,
- 7703BPXA-IF,
- 7703BPX50-IF,
- 7703BPX-LB-DC,

- 7780BPX-DS3,
- 7703BPX-DC-RF

The driver was tested with the following cards:

- Model: 7703BPXA-IF
- SW: 1.2.5
- Model: 7703BPXA-LB
- SW: 1.2.5

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the rf level and threshold of channel A and B and the redundancy parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- operationMode --- Defines the switch behaviour. Has to be set initially as it not read from device.
  - SW-ONCE: Card switches only from channel A to B, Switching back has to be done manually.

    - SW-ALWAYS: Card switch also back to channel A when in respective fault condition.
- delay --- Defines the switching delay in ms
- debounce --- Defines the switch debounce in ms
- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.operationMode	CHOICE	SETUP	EnumRange (SW-ONCE SW-ALWAYS)
config.delay	INTEGER	SETUP	IntegerRange (0 .. 5000)
config.debounce	INTEGER	SETUP	IntegerRange (0 .. 5000)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
position	CHOICE		EnumRange (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
rx.powerA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.powerB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.thresholdA	INTEGER		IntegerRange (-50 .. -10)
rx.thresholdB	INTEGER		IntegerRange (-50 .. -10)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power Ch. A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Ch. B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)
card.name	TEXT	SAVE	StringRange
internal.model	TEXT		StringRange

9.235 Evertz-7707-ET

This page describes the device driver and the device window for the Evertz 7800 Chassis ET series Ethernet transmitter cards of type:

- 7707-ET,
- 7707-ET4,
- 7707ET-TE1

The driver was tested with a 4port card:

- Model: 7707ET4-F2
- SW: 2.3.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the ethernet port states and G703 settings for T1 cards
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
rxErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rxPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
port.01.mode	CHOICE	R/O	EnumRange R/O (HALF DUPLEX FULL DUPLEX)
port.01.speed	CHOICE	R/O	EnumRange R/O (10M 100M 1000M)

name	*type*	*flags*	*range*
port.01.operState	CHOICE	R/O	EnumRange R/O (LINK DOWN LINK UP)
port.02.mode	CHOICE	R/O	EnumRange R/O (HALF DUPLEX FULL DUPLEX)
port.02.speed	CHOICE	R/O	EnumRange R/O (10M 100M 1000M)
port.02.operState	CHOICE	R/O	EnumRange R/O (LINK DOWN LINK UP)
port.03.mode	CHOICE	R/O	EnumRange R/O (HALF DUPLEX FULL DUPLEX)
port.03.speed	CHOICE	R/O	EnumRange R/O (10M 100M 1000M)
port.03.operState	CHOICE	R/O	EnumRange R/O (LINK DOWN LINK UP)
port.04.mode	CHOICE	R/O	EnumRange R/O (HALF DUPLEX FULL DUPLEX)
port.04.speed	CHOICE	R/O	EnumRange R/O (10M 100M 1000M)
port.04.operState	CHOICE	R/O	EnumRange R/O (LINK DOWN LINK UP)
g703.mode	CHOICE	R/O	EnumRange R/O DIS (T1 E1 J1)
g703.state	CHOICE	R/O	EnumRange R/O DIS (ACTIVE NOT ACTIVE)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Laser)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Link Status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Port#1 Link Down)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Port#2 Link Down)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Port#3 Link Down)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Port#4 Link Down)
internal.model	TEXT		StringRange

9.236 Evertz-7707-LR

This page describes the device driver and the device window for the Evertz 7800 Chassis

LR7707 Series 70/140MHz Receiver cards type:

- 7707LR,
- 7707LRH,
- 7707IFR,
- 7707IFRH,
- 7707LRW,
- 7707IFRA,
- 7707FRAH

The driver was tested with the following cards:

- Model: 7707IFRA
- SW: 0.3.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all parameters of the LR card.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- squelch --- Defines if squelch is activated or not

- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.squelch	CHOICE	SETUP	EnumRange (ON OFF)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
optLowThreshold	INTEGER		IntegerRange (-33 .. 3)
optHighThreshold	INTEGER		IntegerRange (-33 .. 4)
rfLowThreshold	INTEGER		IntegerRange (-35 .. -4)
rfHighThreshold	INTEGER		IntegerRange (-35 .. -4)
optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
partnerTxPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.gain	CHOICE		EnumRange (AUTO 10dB 20dB 30dB)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Opt Pwr Low)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Opt Pwr High)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF Out Low)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Out High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
card.name	TEXT	SAVE	StringRange
internal.model	TEXT		StringRange

9.237 Evertz-7707-LT

This page describes the device driver and the device window for the Evertz 7800 Chassis LT7707 Series 70/140MHz transmitter cards type:

- 7707LT,
- 7707LTA,
- 7707IFT,
- 7707LTAW,
- 7707IFTA

The driver was tested with the following cards:

- Model: 7707IFTA
- SW: 0.2.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all parameters of the LT card.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- lnbVoltage --- Defines the lnb supply voltage for suitable cards
- lnb --- Enable lnb supply for suitable cards.
- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lnbVoltage	CHOICE	R/O	EnumRange R/O (OFF LOW HIGH)
config.lnb	CHOICE	R/O	EnumRange R/O (ON OFF)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
optCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rfLowThreshold	INTEGER		IntegerRange (0 .. 0)
rfHighThreshold	INTEGER		IntegerRange (0 .. 0)
tx.gain	CHOICE		EnumRange (AUTO 0dB 10dB 15dB)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Laser Failed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Laser Degraded)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF In Low)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF In High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
card.name	TEXT	SAVE	StringRange
internal.model	TEXT		StringRange

9.238 Evertz-7708-LR

This page describes the device driver and the device window for the Evertz 7800 Chassis LR7708 Series Wideband Receiver cards type:

- 7708LRA,
- 7708LR-H,
- 7708-LR

The driver was tested with the following cards:

- Model: 7708LRA
- SW: 1.12 Build 120

Please note: Parameters of the corresponding transmit card are not implemented. Use separate 7708-LT driver on the opposite chassis.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all parameters of the LR card.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- agcMode --- Defines the gain mode of the card:
    - AGC: Maintains a constant output level even if the rf input level changes as long as in AGCs hold range.
    - IGC: Compensates optical loss and outputs rf signal proportional to what is coming into the transmitter card.
    - MANUAL: User selectable fixed gain for rf levels.
- agcLevel --- Defines the target output level when in AGC mode.
- squelch --- Defines if squelch is activated or not
- squelchThreshold --- Defines the threshold for transmitters cards rf input level. Works only with compatible cards which send monitoring information to receive cards.
- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.agcMode	CHOICE	SETUP	EnumRange (AGC IGC MANUAL)
config.agcLevel	INTEGER	SETUP	IntegerRange (-40 .. -10)
config.squelch	CHOICE	SETUP	EnumRange (ON OFF)
config.squelchThreshold	INTEGER	SETUP	IntegerRange (-65 .. -40)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
optLowThreshold	INTEGER		IntegerRange (-40 .. -4)
optHighThreshold	INTEGER		IntegerRange (-3 .. 5)
rfLowThreshold	INTEGER		IntegerRange (-55 .. -30)
rfHighThreshold	INTEGER		IntegerRange (-29 .. 0)
rx.gain	FLOAT		DoubleRange (-20.0 .. 50.0)
rx.squelch	CHOICE	R/O	EnumRange R/O (DISABLED ACTIVE NOT ACTIVE)
optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Opt Pwr Low)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Opt Pwr High)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF Out Low)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Out High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AGC)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Squelch)
internal.model	TEXT		StringRange
card.name	TEXT	SAVE	StringRange

9.239 Evertz-7708-LT

This page describes the device driver and the device window for the Evertz 7800 Chassis LT7708 Series Wideband transmitter cards type:

- 7708LT,
- 7708LT-LNB,
- 7708LT-DWDM,
- 7708LT-DWDM-LNB

The driver was tested with the following cards:

- Model: 7708LT-LNB
- SW: 1.5 build73

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all parameters of the LT card.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- agcMode --- Defines the gain mode of the card:
  - AGC: Maintains a constant output level even if the rf input level changes as long as in AGCs hold range.
  - MANUAL: User selectable fixed gain for rf levels.
- agcLevel --- Defines the target output level when in AGC mode.
- squelch --- Defines if output squelch is activated or not
- squelchThreshold --- Defines the transmitter input level threshold for enabling output squelch.
- lnbVoltage --- Defines the lnb supply voltage for suitable cards
- lnbTone --- Enabling the 22kHZ tone for suitable cards.
- lnbLowThreshold --- Defines the low lnb current level threshold for suitable cards.
- lnbHighThreshold --- Defines the high lnb current level threshold for suitable cards.
- slot --- Defines the slot number where the card is installed. Setting this parameter is indispensable for correct communication

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.agcMode	CHOICE	SETUP	EnumRange (AGC MANUAL)
config.agcLevel	INTEGER	SETUP	IntegerRange (-50 .. 0)
config.squelch	CHOICE	SETUP	EnumRange (ON OFF)
config.squelchThreshold	INTEGER	SETUP	IntegerRange (-65 .. -40)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.lnbTone	CHOICE	SETUP	EnumRange (ON OFF)
config.lnbLowThreshold	INTEGER	SETUP	IntegerRange (0 .. 500)
config.lnbHighThreshold	INTEGER	SETUP	IntegerRange (0 .. 500)
config.slot	CHOICE	SAVE SETUP	EnumRange (2 3 4 5 6 7 8 9 10 11 12 13 14 15)
rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rfLowThreshold	INTEGER		IntegerRange (0 .. 0)
rfHighThreshold	INTEGER		IntegerRange (0 .. 0)
tx.gain	FLOAT		DoubleRange (0.0 .. 63.0)
tx.squelch	CHOICE	R/O	EnumRange R/O (DISABLED ACTIVE NOT ACTIVE)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Laser Failed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Output Squelch)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF In Low)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF In High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AGC)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (LNB Short)
faults.07	ALARM	R/O	AlarmFlagRange R/O (LNB Current Low)
faults.08	ALARM	R/O	AlarmFlagRange R/O (LNB Current High)
card.name	TEXT	SAVE	StringRange
internal.model	TEXT	R/O	StringRange R/O

9.240 Evertz-7800-Frame

This page describes the device driver and the device window for the Evertz 7800FC Main Frame with controller via SNMP.

The installed cards are using separated device drivers and should be installed in the same thread in the vlc setup. The main frame driver gives an overview about the installed cards and the frames power supplies, fans and temperature status.

The driver was tested with the following SW Releases:

- Kernel: 5.1 2014-11-24
- SW: 1.3 Build 18 2015-01-20 (15-Card)
- SW: 1.3.Build 21 2015-08-13 (4-Card)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Overview of installed cards Slot 2-15
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification

consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- numberOfSlots --- Defines if the unit is a 4 or 15 card controller frame

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.kernel	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.numberOfSlots	CHOICE	SAVE SETUP	EnumRange (4 15)

name	*type*	*flags*	*range*
slot.01.card	TEXT	R/O	StringRange R/O
slot.02.card	TEXT	R/O	StringRange R/O
slot.03.card	TEXT	R/O	StringRange R/O
slot.04.card	TEXT	R/O	StringRange R/O
slot.05.card	TEXT	R/O	StringRange R/O
slot.06.card	TEXT	R/O	StringRange R/O
slot.07.card	TEXT	R/O	StringRange R/O
slot.08.card	TEXT	R/O	StringRange R/O
slot.09.card	TEXT	R/O	StringRange R/O
slot.10.card	TEXT	R/O	StringRange R/O
slot.11.card	TEXT	R/O	StringRange R/O
slot.12.card	TEXT	R/O	StringRange R/O
slot.13.card	TEXT	R/O	StringRange R/O
slot.14.card	TEXT	R/O	StringRange R/O
slot.15.card	TEXT	R/O	StringRange R/O
slot.01.name	TEXT	SAVE	StringRange
slot.02.name	TEXT	SAVE	StringRange
slot.03.name	TEXT	SAVE	StringRange
slot.04.name	TEXT	SAVE	StringRange
slot.05.name	TEXT	SAVE	StringRange
slot.06.name	TEXT	SAVE	StringRange
slot.07.name	TEXT	SAVE	StringRange
slot.08.name	TEXT	SAVE	StringRange
slot.09.name	TEXT	SAVE	StringRange
slot.10.name	TEXT	SAVE	StringRange
slot.11.name	TEXT	SAVE	StringRange
slot.12.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
slot.13.name	TEXT	SAVE	StringRange
slot.14.name	TEXT	SAVE	StringRange
slot.15.name	TEXT	SAVE	StringRange
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Frame Status Line)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
internal.oid	INTEGER		IntegerRange (0 .. 0)

9.241 Evertz-7881-IRD

This page describes the device driver and the device window for the Evertz 7881 Mini IRD in a modular frame controlled via its RP network port not through the Frame Controller via SNMP.

Please Note: Units LO Frequency cannot be set to 0. Therefore useLoFreq has to be set to NO on setup page to be able to operate the driver together with sat-nms channel database

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the Program, service list and sat receiver settings and status display.
- --- Contains IP in / out parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the

audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- lnbVoltage --- Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

- bandEdge --- The frequency where the software switches the LO frequencies.
- serviceHunt --- This parameter determines if the IRD will automatically choose a server / set of PIDs when tuned to a new frequency. With the Ateme IRD, this parameter also selects one of three basic operation modes for PID selection (see paragraph "PID selection" below).
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- marginOffset --- Defines an offset which will be added to the margin shown to the operator.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
- useLoFreq --- If set to YES, the driver reads back the units set LO frequency and adds or recalculates L-Band frequency to be compatible with satnms CDB.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.marginOffset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.useLoFreq	CHOICE	SAVE SETUP	EnumRange (YES NO)
internal.band	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
internal.fpsn	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.fpsd	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.margin	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.interface	CHOICE	R/O	EnumRange R/O (IP ASI SAT)
internal.input	CHOICE	R/O	EnumRange R/O (IP ASI SAT)
internal.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bissmode	CHOICE	R/O	EnumRange R/O (OFF BISS-1 BISS-E BISS-E1 BISS-E2)
rfid	CHOICE	R/O	EnumRange R/O (1 2 3 4)
reset	TEXT	NOPRESET	StringRange
interface	CHOICE		EnumRange (IP ASI SAT)
input	CHOICE		EnumRange (1 2 3 4)
frequency	INTEGER		IntegerRange (0 .. 0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK AUTO)
state.modulation	CHOICE	R/O	EnumRange R/O (QPSK 8PSK 16APSK 32APSK UNKNOWN)
fec	TEXT	R/O	StringRange R/O
rolloff	CHOICE		EnumRange (20% 25% 35%)
searchRange	INTEGER		IntegerRange (0 .. 67500)
polarization	CHOICE		EnumRange (X Y)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
serviceMode	CHOICE		EnumRange (AUTOMATIC PIDLOCKED)

name	*type*	*flags*	*range*
internal.serviceMode	CHOICE	R/O	EnumRange R/O (AUTOMATIC PIDLOCKED)
output.conn3	CHOICE		EnumRange (VIDEO ASI)
video.pid	INTEGER		IntegerRange (0 .. 0)
video.framesync	CHOICE		EnumRange (EXTERNAL INTERNAL)
video.fail	CHOICE		EnumRange (FREEZE RED BLUE BLACK OFF)
video.latency	CHOICE		EnumRange (ON OFF MIN)
video.delay	INTEGER		IntegerRange (0 .. 200)
video.profile	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
video.resolution	TEXT	R/O	StringRange R/O
video.format	CHOICE		EnumRange (AUTO SD-525 SD-625 HD-1920x1080i-50Hz HD-1920x1080i-59.94Hz HD-1920x1080i-60Hz HD-1280x720p-50Hz HD-1280x720p-59.94Hz HD-1280x720p-60Hz HD-1920x1080p-25Hz HD-1920x1080p-29.97Hz HD-1920x1080p-30Hz HD-1920x1080p-50Hz HD-1920x1080p-59.94Hz HD-1920x1080p-60Hz)
audio.mode	CHOICE		EnumRange (AUTO PCM DOLBY-E AC3)
audio.1.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.1.delay	INTEGER		IntegerRange (0 .. 200)
audio.1.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.1.pid	INTEGER		IntegerRange (0 .. 8190)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.2.delay	INTEGER		IntegerRange (0 .. 200)
audio.2.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.2.pid	INTEGER		IntegerRange (0 .. 8190)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.3.delay	INTEGER		IntegerRange (0 .. 200)
audio.3.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.3.pid	INTEGER		IntegerRange (0 .. 8190)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.4.delay	INTEGER		IntegerRange (0 .. 200)
audio.4.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.4.pid	INTEGER		IntegerRange (0 .. 8190)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.5.delay	INTEGER		IntegerRange (0 .. 200)
audio.5.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.5.pid	INTEGER		IntegerRange (0 .. 8190)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.6.delay	INTEGER		IntegerRange (0 .. 200)
audio.6.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.6.pid	INTEGER		IntegerRange (0 .. 8190)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.7.delay	INTEGER		IntegerRange (0 .. 200)
audio.7.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.7.pid	INTEGER		IntegerRange (0 .. 8190)
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.mode	CHOICE		EnumRange (DECODE BYPASS)
audio.8.delay	INTEGER		IntegerRange (0 .. 200)

name	*type*	*flags*	*range*
audio.8.downmix	CHOICE		EnumRange (DOWNMIX FULL)
audio.8.pid	INTEGER		IntegerRange (0 .. 8190)
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.multiProgram	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.snr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.ebno	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.margin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dvbmode	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E1 BISS-E2)
ca.bisskey	TEXT		StringRange
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (0 .. 0)
ipin.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.bufferDuration	INTEGER		IntegerRange (1 .. 1000)
ipin.reset	TEXT	NOPRESET	StringRange
ipin.type	CHOICE	R/O	EnumRange R/O (UDP RTP)

name	*type*	*flags*	*range*
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.fecParity	CHOICE	R/O	EnumRange R/O (1D 2D)
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.bufferDuration	INTEGER	R/O	IntegerRange R/O (1 .. 1000)
ipout.enable	CHOICE		EnumRange (OFF ON)
ipout.address	TEXT		StringRange
ipout.port	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.242 Evertz-EMR-Matrix

This page describes the device driver and the device window for the Evertz EMR series distributive switch matrix/router using the Quartz Routing Switcher Remote Control Protocol.

The List Routes command .L{level}[dest},- [cr} is not working on the tested unit and therefore is not implemented

The driver was developed and tested with EMR-64x64-3G/6G/12G (64x64 SD/HD/3G/6G/12G Modular Router)

- Software Build 0715 Rev.10
- MPF Version: 4.00051

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port Switching by alias name outputs 1-32.
- abc --- Port Switching by alias name outputs 33-64.
- 123 --- Port Switching by number outputs 1-32.
- 123 --- Port Switching by number outputs 33-64.
- --- Input alias names.
- --- Output alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters for the Evertz Maztrix.

parameter	***description***
address	N/A
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.readOnce	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
aliasRead	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.16.name	TEXT		StringRange
output.17.name	TEXT		StringRange
output.18.name	TEXT		StringRange
output.19.name	TEXT		StringRange
output.20.name	TEXT		StringRange
output.21.name	TEXT		StringRange
output.22.name	TEXT		StringRange
output.23.name	TEXT		StringRange
output.24.name	TEXT		StringRange
output.25.name	TEXT		StringRange
output.26.name	TEXT		StringRange
output.27.name	TEXT		StringRange
output.28.name	TEXT		StringRange
output.29.name	TEXT		StringRange
output.30.name	TEXT		StringRange
output.31.name	TEXT		StringRange
output.32.name	TEXT		StringRange
output.33.name	TEXT		StringRange
output.34.name	TEXT		StringRange
output.35.name	TEXT		StringRange
output.36.name	TEXT		StringRange
output.37.name	TEXT		StringRange
output.38.name	TEXT		StringRange
output.39.name	TEXT		StringRange
output.40.name	TEXT		StringRange
output.41.name	TEXT		StringRange
output.42.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.43.name	TEXT		StringRange
output.44.name	TEXT		StringRange
output.45.name	TEXT		StringRange
output.46.name	TEXT		StringRange
output.47.name	TEXT		StringRange
output.48.name	TEXT		StringRange
output.49.name	TEXT		StringRange
output.50.name	TEXT		StringRange
output.51.name	TEXT		StringRange
output.52.name	TEXT		StringRange
output.53.name	TEXT		StringRange
output.54.name	TEXT		StringRange
output.55.name	TEXT		StringRange
output.56.name	TEXT		StringRange
output.57.name	TEXT		StringRange
output.58.name	TEXT		StringRange
output.59.name	TEXT		StringRange
output.60.name	TEXT		StringRange
output.61.name	TEXT		StringRange
output.62.name	TEXT		StringRange
output.63.name	TEXT		StringRange
output.64.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
input.17.name	TEXT		StringRange
input.18.name	TEXT		StringRange
input.19.name	TEXT		StringRange
input.20.name	TEXT		StringRange
input.21.name	TEXT		StringRange
input.22.name	TEXT		StringRange
input.23.name	TEXT		StringRange
input.24.name	TEXT		StringRange
input.25.name	TEXT		StringRange
input.26.name	TEXT		StringRange
input.27.name	TEXT		StringRange
input.28.name	TEXT		StringRange
input.29.name	TEXT		StringRange
input.30.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.31.name	TEXT		StringRange
input.32.name	TEXT		StringRange
input.33.name	TEXT		StringRange
input.34.name	TEXT		StringRange
input.35.name	TEXT		StringRange
input.36.name	TEXT		StringRange
input.37.name	TEXT		StringRange
input.38.name	TEXT		StringRange
input.39.name	TEXT		StringRange
input.40.name	TEXT		StringRange
input.41.name	TEXT		StringRange
input.42.name	TEXT		StringRange
input.43.name	TEXT		StringRange
input.44.name	TEXT		StringRange
input.45.name	TEXT		StringRange
input.46.name	TEXT		StringRange
input.47.name	TEXT		StringRange
input.48.name	TEXT		StringRange
input.49.name	TEXT		StringRange
input.50.name	TEXT		StringRange
input.51.name	TEXT		StringRange
input.52.name	TEXT		StringRange
input.53.name	TEXT		StringRange
input.54.name	TEXT		StringRange
input.55.name	TEXT		StringRange
input.56.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.57.name	TEXT		StringRange
input.58.name	TEXT		StringRange
input.59.name	TEXT		StringRange
input.60.name	TEXT		StringRange
input.61.name	TEXT		StringRange
input.62.name	TEXT		StringRange
input.63.name	TEXT		StringRange
input.64.name	TEXT		StringRange
output.01	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.02	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.03	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.04	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.05	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.06	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.07	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.08	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.09	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.10	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.11	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.12	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.13	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.14	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.15	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.16	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.17	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.18	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.19	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.20	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.21	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.22	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.23	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.24	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.25	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.26	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.27	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.28	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.29	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.30	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.31	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.32	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.33	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.34	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.35	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.36	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.37	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.38	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.39	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.40	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.41	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.42	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.43	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.44	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.45	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.46	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.47	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.48	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.49	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.50	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.51	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.52	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.53	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.54	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.55	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.56	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.57	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.58	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.59	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.60	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.61	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.62	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
output.63	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)

name	*type*	*flags*	*range*
output.64	CHOICE		EnumRange (001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)

9.243 Execute

This logical device executes a script every time a parameter changes and submits this parameter to the script.

Typical application is to react on the summary fault and trigger a local script or program on the MNC server.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	***description***
script	The name of the script which should be executed
inputId	Enter the message identifier here of the parameter the device shall listen

Security advice and Remarks

The Execute device runs programs and scripts of the Linux operating system which are located in /home/satnms/scripts/.

Using the Execute device may be dangerous, if the executed script does not behave properly and if it does not handle input values in a proper way.

1. Scripts may be scripts requiring bash, perl etc and Linux executables as well
2. Scripts and programs must be locates in the /home/satnms/scripts/ subdirectory, they must be readable / executable to the 'satnms' user.
3. The software reads the output of the program to stdout but ignores the stderr output. This is useful for some Linux programs which send their output to stdout and status reports to stderr. If the stderrr output is required, the program should be enclosed in a bash script performing the appropriate redirection.
4. There are no timeouts with the Script protocol. If a script or program doesn't terminate, it blocks the device thread containing the driver forever without notice for the operator. When the MNC server terminates, the subprocess is forced to terminate as well.
5. There is no explicit limitation of the volume of data collected from the script's output. Hence, an endless running program which continuously output e.g. some logging data, will eat up the complete memory of the MNC server process an bring the system down.

Be careful when using the Execute device!

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.lastMessage	TEXT	R/O	StringRange R/O
config.script	TEXT	SAVE SETUP	StringRange
config.inputId	TEXT	SAVE SETUP	StringRange
input	TEXT	NOPRESET	StringRange
run	TEXT	NOPRESET	StringRange
arguments	TEXT	SAVE NOPRESET	StringRange

9.244 Experimental

The Experimental driver is a stub for own driver experiments. It is not a functional device driver. Be aware, that this file gets overwritten with each software update you apply to the sat-nms installation.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.load	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
address	TEXT		StringRange

name	*type*	*flags*	*range*
position	TEXT	SAVE	StringRange
xval	HEX		HexRange (0 .. FF)

9.245 Fanuc-PLC9030

This page describes the device driver and the device window for the*Fanuc-PLC9030* device. This device controls up to 320 inputs and up to 320 outputs at a Fanuc Series 90 / 30 PLC. It communicates to the PLC through a CCM (Communication Coprocessor Module) and requires the PLC to be configured in a special way (see the paragraph at the end of this chapter).

The *Fanuc-PLC9030* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Fanuc-PLC9030* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the*Fanuc-PLC9030* device driver as a low level driver to access the physical I/O ports.

Port addressing

The device driver gives access to the I/O ports through five integer variables each for inputs and for outputs. Each variable stands for 64 input or output ports, which in total results in a pool of 320 input and 320 output ports.

The variables are named in way, that a translation of a particular port to a bit position in the variable is quite easy. Example:

An input port which appears in the Fanuc PLC as '%I00066' is accessed through the variable I00065 of the device driver. It is represented by the bit 2^1 in the variable (the least significant bit represents the '%I00065' input).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Fanuc PLC.

parameter	***description***
address	Leave this field empty, this driver uses a fixed "CCM CPU ID" (see below)
inputPorts	Select the number of input ports the PLC is configured to use.
outputPorts	Select the number of output ports the PLC is configured to use.
Serial	Remember to set the line parameters for the serial interface connected to the Fanuc PLC. The setting required by the Fanuc is *19200,O,8,1* .

Configuring the Fanuc PLC for the usage with this driver

To use a Fanuc 90/30 PLC as an I/O module for the sat-nms software, the PLC must be equipped with a CCM (Communication Coprocessor Module). The sat-nms software does not support a communication through the CPU serial interface.

The PLC must be configured in the following way (using a hand held controller ot the Fanuc PC software):

- Each I/O module must be placed in the input resp. output table at the lowest possible address. There must not be gaps of unused port addresses in the input/output tables.
- The total number of configures I/O ports must be set at the configuration page of the Fanuc-PLC9030 device in the sat-nms software (configuration parameters 'inputPorts' and 'outputPorts'.
- If the PLC is running a program, that actively sets or clears any output ports, the sat-nms software will not recognize these changes. The software always will signal the setting which recently was commanded. The state of outputs in the PLC is read back only after power on / after communication loss.
- The CCM must be configured to the following settings:
  - Configuration mode --- CCM ONLY
  - CCM Enable --- YES
  - CCM Mode --- SLAVE
  - Interface --- RS232
  - Data Rate --- 19200
  - Flow Control --- NONE
  - Parity --- ODD
  - Retry Count --- NORMAL
  - Timeout --- LONG
  - Modem ... --- NONE
  - CCM CPI ID --- 1

The required settings are the factory settings of the CCM as described in the Fanuc "Series 90 PLC Serial Communications User's Manual".

Variables defined by this device driver

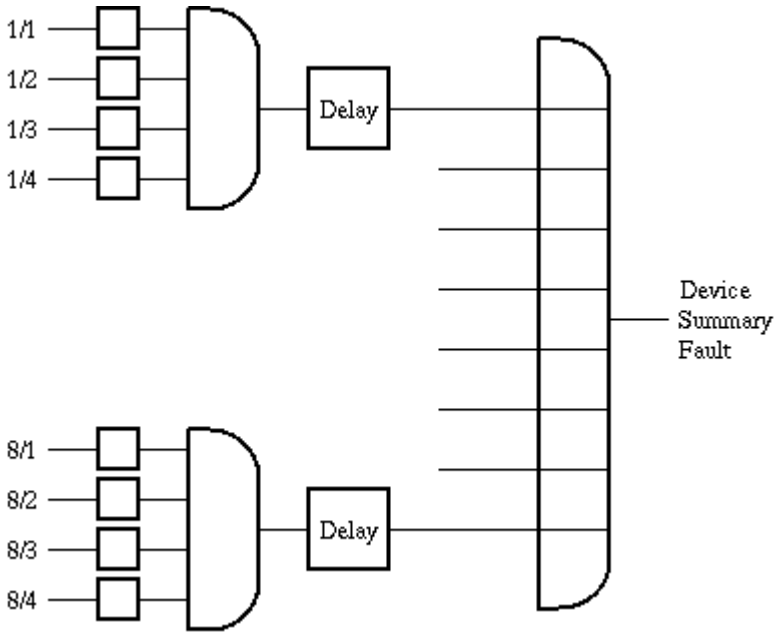
name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.dummy	HEX	R/O	HexRange R/O (0 .. 0)
config.inputPorts	CHOICE	SAVE SETUP	EnumRange (0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320)
config.outputPorts	CHOICE	SAVE SETUP	EnumRange (0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320)
I00001	HEX	R/O	HexRange R/O (0 .. 0)
I00065	HEX	R/O	HexRange R/O (0 .. 0)
I00129	HEX	R/O	HexRange R/O (0 .. 0)
I00193	HEX	R/O	HexRange R/O (0 .. 0)
I00257	HEX	R/O	HexRange R/O (0 .. 0)
Q00001	HEX		HexRange (0 .. 0)
Q00065	HEX		HexRange (0 .. 0)
Q00129	HEX		HexRange (0 .. 0)
Q00193	HEX		HexRange (0 .. 0)
Q00257	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

9.246 Fault-Array

The Fault-Array logical device combines fault messages or other state parameters of arbitrary

devices in a flexible way to one common fault state. Comparable to a programmable gate array, it combines the input information on a user configurable basis.

The Fault-Array contains so called elements in 8 rows and 4 columns. Each element can be configured via parameter messages to listen to a certain parameter value. The element owns a FAULT state, which becomes true if the parameter value matches a target value. This fault state is generated by comparing the parameter value to a user defined target. Thus the Fault-Array is capable to process arbitrary parameters like switch positions or "TX-ON" settings beside the fault state of other devices.



The fault states of the elements in one row are ANDed together (i.e. the row summary fault becomes true if all elements are faulty at a time). The row summary fault is passed though some sort of "glitch filter" which suppresses faults shorter than a certain time. All filtered row fault states are ORed to the device summary fault flag.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- $_{1}$ ... $_{8}$ --- The device window provides eight pages, each containing the parameters of one table row. The meaning of each particular parameter is explained below.
- --- The device info page.
- --- The maintenance page.

Setting up a Fault-Array logical device

The most common use of a Fault-Array logical device will be to combine several fault flags to the switch condition of a redundancy switch or a site diversity switch.

The first eight pages of the device window show the parameters of one row (AND gate) of the Fault-Array. The table below explains these parameters:

parameter	***description***
message id	You specify the source of data you want to route to the gate input with this field. A message identifier is made of the name of the device followed by a dot and the name of the parameter which shall be fed into this input. The fault flag "faults.05" of the device "REC-1" e.g. uses the message id "REC-1.faults.05". A complete list of parameter names for all device types is found later in this document. Leave this field empty if the input shall not be connected.
actual value	This read only field shows the actual value of the parameter connected to this gate input.
target value	The actual parameter value gets compared to this target value in order to set the gate input. Be aware that this is a textual comparison, FLOAT parameters may fail to match the target value due to rounding errors.
comparison	If "==" is selected here, the gate input becomes true ("FAULT") if the parameter value matches the target. The selection of "!=" reverses this.
momentary	This read only field shows the momentary output state of the row/gate before it gets processed by the delay.
delay	A delay time may specified here to filter out short-time events which must not appear in the overall output state. The momentary output of the gate must be in "FAULT" state for at least this delay time before the device summary fault is raised.
filtered	This read only field shows the filtered output state of the row/gate after it has been processed by the delay.

Remarks

The filter/delay function is implemented by the background polling procedure which is called on a regular basis for all logical devices. With the factory preset settings, this procedure runs once a second. The resolution of the delay time is limited to this value.

If your application needs a finer resolution, you may configure the polling procedures of all logical device to be called more frequently. This is done by configuring the *IDLE* parameter of the *PORT null* in the vlc.setup file. Be aware, that the workload of the M&C computer may rise dramatically if you set this parameter to very small values.

As of version 1.01 (2014-04-08), the device sets an alarm flag**faults.01** in parallel to the device summary fault. The **faults.01** flag duplicates the the summary fault, neither the priority nor the delay time of this flag may be configured by the operator.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Overall output)

9.247 Fault-Gate-3

The Fault-Gate-3 logical device monitors three faults flags from other devices and creates a summary warning or alarm depending on the number of fault flags which are active or not.

The device has 4 return states:

- OK = all three faults are off / no faults
- WARNING = one or two fault flags are active
- ALARM = all three faults are active
- UNDEF = device not initialized yet

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page.

Configuration

parameter	*description*
input-A.id	parameter ID first fault
input-B.id	parameter ID second fault
input-C.id	parameter ID third fault
polarity	NORMAL = true means fault, INVERTED = false means fault
config.warnMsg	text of summary warning (will also used for the event log entry)
config.faultMsg	text of summary fault (will also used for the event log entry)
enable-A.Id, enable-A.Value	use the first parameter to calculate the summary warning and alarm only of the enable-A.Id has the value enable-A.Value

parameter	*description*
enable-B.Id, enable-B.Value	use the second parameter to calculate the summary warning and alarm only of the enable-B.Id has the value enable-B.Value
enable-C.Id, enable-C.Value	use the third parameter to calculate the summary warning and alarm only of the enable-C.Id has the value enable-C.Value

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.input-C.Id	TEXT	SAVE SETUP	StringRange
config.polarity	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERTED)
config.warnMsg	TEXT	SAVE SETUP	StringRange
config.faultMsg	TEXT	SAVE SETUP	StringRange
config.enable-A.Id	TEXT	SAVE SETUP	StringRange
config.enable-A.Value	TEXT	SAVE SETUP	StringRange
config.enable-B.Id	TEXT	SAVE SETUP	StringRange
config.enable-B.Value	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.enable-C.Id	TEXT	SAVE SETUP	StringRange
config.enable-C.Value	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (UNDEF OK WARNING ALARM)
internal.a	BOOLEAN		BooleanRange (true false)
internal.b	BOOLEAN		BooleanRange (true false)
internal.c	BOOLEAN		BooleanRange (true false)
internal.i	INTEGER		IntegerRange (0 .. 0)
internal.dis	INTEGER		IntegerRange (0 .. 0)
internal.ea	TEXT		StringRange
internal.eb	TEXT		StringRange
internal.ec	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Warning)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Alarm)

9.248 Fault-Gate

The Fault-Gate logical device monitors two faults flags from other devices and creates a summary warning or alarm depending on the number of fault flags which are active or not.

The device has 4 return states:

- OK = all two faults are off / no faults
- WARNING = one fault flags are active
- ALARM = all two faults are active
- UNDEF = device not initialized yet

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page.

Configuration

parameter	*description*
input-A.id	parameter ID first fault
input-B.id	parameter ID second fault
polarity	NORMAL = true means fault, INVERTED = false means fault
config.warnMsg	text of summary warning (will also used for the event log entry)
config.faultMsg	text of summary fault (will also used for the event log entry)

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.polarity	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERTED)
config.warnMsg	TEXT	SAVE SETUP	StringRange
config.faultMsg	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (UNDEF OK WARNING ALARM)
internal.a	BOOLEAN		BooleanRange (true false)
internal.b	BOOLEAN		BooleanRange (true false)
internal.i	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Warning)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Alarm)

9.249 FeiZyfer-GPS-Receiver

This page describes the monitoring only device driver and the device window for the FeiZyfer GPS GSync Synchronized Time and Frequency System / CommSync2 Product Family

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The GPS related parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.partNo	TEXT	R/O	StringRange R/O
info.ssta	TEXT	R/O	StringRange R/O
longitude	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
latitude	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
utcTime	TEXT	R/O	StringRange R/O
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
state	CHOICE	R/O	EnumRange R/O (WARM-UP TIME LOCKED COASTING ACQUIRE ALARM FREQUENCY LOCKED)
faults.01	ALARM	R/O	AlarmFlagRange R/O (GPS receive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power)
faults.03	ALARM	R/O	AlarmFlagRange R/O (10mhz)
faults.04	ALARM	R/O	AlarmFlagRange R/O (GPS communication)
faults.05	ALARM	R/O	AlarmFlagRange R/O (1PPS)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Module not ready)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Antenna over current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Antenna under current)
internal.hour	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minute	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.second	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dummy	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.250 File-Recorder

The File-Recorder logical device provides a simple facility to write measurement or state values in regular intervals to a file. In user configurable intervals, the File-Recorder writes a line consisting of a time stamp and up to 4 measurement values to the file. A file size limit may be defined to prevent the File-Recorder from creating arbitrary sized files.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page contain only one button which launches a History File Viewer to display the recorded data graphically.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the File-Recorder device with the devices which origin the measurement values to record.

- *parameter --- description*
- channel1Id --- The message identifier of the parameter to record as 'channel 1'. If left empty, an empty column is written to the file for this channel.
- channel2Id --- The message identifier of the parameter to record as 'channel 2'. If left empty, an empty column is written to the file for this channel.
- channel3Id --- The message identifier of the parameter to record as 'channel 3'. If left empty, an empty column is written to the file for this channel.
- channel4Id --- The message identifier of the parameter to record as 'channel 4'. If left empty, an empty column is written to the file for this channel.
- interval --- The time mean interval between two lines written to the file. The true interval may vary by about one second. Below some long term intervals, expressed in seconds.
  - 1 hour: --- 3600 seconds
  - 24 hours: --- 86400 seconds
  - 7 days: --- 604800 seconds

    - 30 days: --- 2592000 seconds

- liveInterval --- The time mean interval for updating the live view buffer.
- maxSize --- The maximum file size (KBytes) the recorder will produce. If set to a non-zero value, the File-Recorder backs up the output file to a file named 'XXXX. old.txt' as soon as the size limit is reached. The recent values are written to a new file 'XXXX.txt'. A value of zero disables the file size checking.**Using the File-Recorder device with a small 'interval' value may generate a huge amount of data. Limiting the file size is strongly recommended!**
- directory --- The location where to store the file recorder's data. By default, File-Recorder devices store their files in the base directory of the sat-nms installation. This is /home/satnms/ in most cases. You may define any subdirectoy of this base directory as the location where this File-Recorder shall store its data. The directory already must exist.When you change a File-Recorder's directory, any existing files in the previous directory are moved to the new one.

File Format

The File-Recorder writes the sample to a file 'XXXX.txt', where 'XXXX' is the device name of the File-Recorder device. The file is placed in the Server's base directory unless the 'directory' configuration setting is set to a subdirectory of this.

The file entirely consists of lines, separated by CR/LF pairs. Each line contains a set of measurement values, preceeded by a time stamp. Columns are separated by semicolons, no space characters are inserted between the columns.

Reading A File From Remote

The file written by the File-Recorder (also the backed up data file, if size limiting is enabled) may be read from a remote computer via http. The URL to read the file is:

http://xxx.xxx.xxx.xxx:2007/XXXX.txt or **http://xxx.xxx.xxx.xxx:2007/XXXX. old.txt** respectively.

Hereby 'xxx.xxx.xxx.xxx' is the IP Address or host name of the M&C / VLC server, 'XXXX' is the name of the File-Recorder device on this machine.

If the 'directory' configuration setting is set to e.g. 'somedir', the URLs to read the files are:

http://xxx.xxx.xxx.xxx:2007/somedir/XXXX.txt or
http://xxx.xxx.xxx.xxx:2007/somedir/XXXX.old.txt respectively.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.directory	TEXT	R/O	StringRange R/O
config.channel1Id	TEXT	SAVE SETUP	StringRange
config.channel2Id	TEXT	SAVE SETUP	StringRange
config.channel3Id	TEXT	SAVE SETUP	StringRange
config.channel4Id	TEXT	SAVE SETUP	StringRange
config.interval	INTEGER	SAVE SETUP	IntegerRange (1 .. 2592000)
config.liveInterval	INTEGER	SAVE SETUP	IntegerRange (1 .. 2592000)
config.maxSize	INTEGER	SAVE SETUP	IntegerRange (0 .. 100000)
config.directory	TEXT	SAVE SETUP	StringRange
liveView	TEXT	R/O	StringRange R/O
directory	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Invalid directory)

9.251 Filter

The logical device Filter compares a parameters with a minimum and maximum value end returns the parameter only if it is between min and max. This is similar to band-pass filter. If the value is outside the defined limits, the device will return fixed values as defined in the device setup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the filtered value
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	*description*
input.Id	Parameter ID of the value which should be filtered
min	Minimum / lower limit
max	Maximum / upper limit.
returnMin	value which will be returned if the input is below min
returnMax	value which will be returned if the input is above max
unit	unit of measurement value (free text)

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input.Id	TEXT	SAVE SETUP	StringRange
config.min	TEXT	SAVE SETUP	StringRange
config.max	TEXT	SAVE SETUP	StringRange
config.returnMin	TEXT	SAVE SETUP	StringRange
config.returnMax	TEXT	SAVE SETUP	StringRange
config.unit	TEXT	SAVE SETUP	StringRange
input	TEXT		StringRange
value	TEXT		StringRange
internal.min	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.max	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.v	TEXT	R/O	StringRange R/O

9.252 Floating-Limit

The Floating-Limit logical device checks a value to be within given limits relative to the average of the value calculated over a given number of hours.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual state of the device.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Floating-Limit device with the devices that deliver the input signal and to configure the device's performance.

parameter	***description***
parameterId	The ID of the parameter to watch
averageTime	The number of hours, the device shall average the measurement value as a reference (1..72h)
startupDelay	After system start the device ignores the measurement values for a certain time. Measurement values are not taken into account for the average value during this time and no limit checks are done. The startup delay may be set to 0..120 seconds.
deltaHighLimit	The high limit to be checked. The value is relative to the calculated average of the measurement value. Setting the high limit to e.g. 1.0 makes the device raise an fault if the measurement value raises above its average value + 1.0.
deltaLowLimit	The low limit to be checked. The value is relative to the calculated average of the measurement value and always positive. Setting the low limit to e.g. 1.0 makes the device raise an fault if the measurement value drops below its average value - 1.0.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.parameterId	TEXT	SAVE SETUP	StringRange
config.averageTime	INTEGER	SAVE SETUP	IntegerRange (1 .. 72)
config.startupDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 120)
config.deltaHighLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.deltaLowLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
actualValue	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
averagedValue	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
absLowLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
absHighLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
averagedTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
gauge	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High limit)

9.253 Foxcom-SatLight-Platinum

This page describes the device driver and the device window for the Foxcom optical link chassis SatLight Platinum.

The driver is capable to monitor/control one chassis with up to 12 optical receiver or transmitter modules.

Device window pages

The following table shows which device window pages are available with this individual device

type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances. | **Fault flags**

The driver provides a number of fault flags which reflect the actual state of the unit and its modules. Flags 01 to 12 show the fault state of the modules / slots 1 to 12. A fault is reported for a module which is installed and reads one of the states Minor, Major or Critical.

Faults 13 and 14 report the state of the two power supply modules. Again, Minor, Major or Critical are interpreted as fault, Normal as OK.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.macAddress	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.slot.01.name	TEXT	R/O	StringRange R/O
info.slot.02.name	TEXT	R/O	StringRange R/O
info.slot.03.name	TEXT	R/O	StringRange R/O
info.slot.04.name	TEXT	R/O	StringRange R/O
info.slot.05.name	TEXT	R/O	StringRange R/O
info.slot.06.name	TEXT	R/O	StringRange R/O
info.slot.07.name	TEXT	R/O	StringRange R/O
info.slot.08.name	TEXT	R/O	StringRange R/O
info.slot.09.name	TEXT	R/O	StringRange R/O
info.slot.10.name	TEXT	R/O	StringRange R/O
info.slot.11.name	TEXT	R/O	StringRange R/O
info.slot.12.name	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.slot.02.ch1.rssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.slot.05.ch1.rssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.slot.08.ch1.rssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.slot.11.ch1.rssi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.01.installed	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.01.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.01.name	TEXT	SAVE	StringRange
slot.01.agc	CHOICE		EnumRange (MGC AGC)
slot.01.lnbPower	CHOICE		EnumRange (OFF ON)
slot.01.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.01.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.01.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.01.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.status	TEXT	R/O	StringRange R/O
slot.02.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.02.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX RX1+1 TX1+1 RX2+1 DATA)
slot.02.name	TEXT	SAVE	StringRange
slot.02.agc	CHOICE		EnumRange (MGC AGC)
slot.02.lnbPower	CHOICE		EnumRange (OFF ON)
slot.02.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.02.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.02.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.02.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.02.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.status	TEXT	R/O	StringRange R/O
slot.02.position	CHOICE		EnumRange (1-STBY 2-STBY 3-STBY 1 2)
slot.02.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.02.location	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
slot.02.ch1.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.02.ch1.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.02.ch1.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.chS.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.ch1.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.chS.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.ch2.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.ch2.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.ch2.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.02.ch2.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.03.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.03.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.03.name	TEXT	SAVE	StringRange
slot.03.agc	CHOICE		EnumRange (MGC AGC)

name	*type*	*flags*	*range*
slot.03.lnbPower	CHOICE		EnumRange (OFF ON)
slot.03.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.03.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.03.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.03.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.status	TEXT	R/O	StringRange R/O
slot.04.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.04.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.04.name	TEXT	SAVE	StringRange
slot.04.agc	CHOICE		EnumRange (MGC AGC)
slot.04.lnbPower	CHOICE		EnumRange (OFF ON)
slot.04.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.04.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.04.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.04.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.04.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.status	TEXT	R/O	StringRange R/O
slot.05.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.05.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX RX1+1 TX1+1 RX2+1 DATA)
slot.05.name	TEXT	SAVE	StringRange
slot.05.agc	CHOICE		EnumRange (MGC AGC)
slot.05.lnbPower	CHOICE		EnumRange (OFF ON)
slot.05.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.05.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.05.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.05.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.status	TEXT	R/O	StringRange R/O
slot.05.position	CHOICE		EnumRange (1-STBY 2-STBY 3-STBY 1 2)
slot.05.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.05.location	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
slot.05.ch1.position	CHOICE	R/O	EnumRange R/O (1 2)

name	*type*	*flags*	*range*
slot.05.ch1.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.05.ch1.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.chS.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.ch1.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.chS.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.ch2.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.ch2.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.ch2.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.05.ch2.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.06.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.06.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.06.name	TEXT	SAVE	StringRange
slot.06.agc	CHOICE		EnumRange (MGC AGC)
slot.06.lnbPower	CHOICE		EnumRange (OFF ON)
slot.06.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.06.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.06.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.06.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.06.status	TEXT	R/O	StringRange R/O
slot.07.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.07.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.07.name	TEXT	SAVE	StringRange
slot.07.agc	CHOICE		EnumRange (MGC AGC)
slot.07.lnbPower	CHOICE		EnumRange (OFF ON)
slot.07.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.07.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.07.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.07.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.status	TEXT	R/O	StringRange R/O
slot.08.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.08.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX RX1+1 TX1+1 RX2+1 DATA)
slot.08.name	TEXT	SAVE	StringRange
slot.08.agc	CHOICE		EnumRange (MGC AGC)
slot.08.lnbPower	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot.08.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.08.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.08.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.08.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.status	TEXT	R/O	StringRange R/O
slot.08.position	CHOICE		EnumRange (1-STBY 2-STBY 3-STBY 1 2)
slot.08.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.08.location	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
slot.08.ch1.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.08.ch1.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.08.ch1.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.chS.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.ch1.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.chS.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.ch2.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.ch2.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.ch2.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.08.ch2.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.09.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)

name	*type*	*flags*	*range*
slot.09.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.09.name	TEXT	SAVE	StringRange
slot.09.agc	CHOICE		EnumRange (MGC AGC)
slot.09.lnbPower	CHOICE		EnumRange (OFF ON)
slot.09.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.09.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.09.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.09.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.09.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.09.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.09.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.status	TEXT	R/O	StringRange R/O
slot.10.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.10.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.10.name	TEXT	SAVE	StringRange
slot.10.agc	CHOICE		EnumRange (MGC AGC)
slot.10.lnbPower	CHOICE		EnumRange (OFF ON)
slot.10.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.10.reqRfOut	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
slot.10.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.10.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.10.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.10.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.10.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.status	TEXT	R/O	StringRange R/O
slot.11.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.11.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.11.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX RX1+1 TX1+1 RX2+1 DATA)
slot.11.name	TEXT	SAVE	StringRange
slot.11.agc	CHOICE		EnumRange (MGC AGC)
slot.11.lnbPower	CHOICE		EnumRange (OFF ON)
slot.11.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.11.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.11.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.11.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.11.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.11.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.11.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.11.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.11.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.11.status	TEXT	R/O	StringRange R/O
slot.11.position	CHOICE		EnumRange (1-STBY 2-STBY 3-STBY 1 2)
slot.11.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.11.location	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
slot.11.ch1.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.11.ch1.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.11.ch1.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.chS.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.ch1.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.chS.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.ch2.rssi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.ch2.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.ch2.position	CHOICE	R/O	EnumRange R/O (1 2)
slot.11.ch2.protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
slot.12.mode	CHOICE	SAVE SETUP	EnumRange (OPERATIONAL MAINTENANCE)
slot.12.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.card	CHOICE	R/O	EnumRange R/O (UNKNOWN NONE TX RX)
slot.12.name	TEXT	SAVE	StringRange
slot.12.agc	CHOICE		EnumRange (MGC AGC)
slot.12.lnbPower	CHOICE		EnumRange (OFF ON)
slot.12.lnbVoltage	CHOICE		EnumRange (13V 18V)
slot.12.reqRfOut	INTEGER		IntegerRange (0 .. 0)
slot.12.reqImd	INTEGER		IntegerRange (0 .. 0)
slot.12.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.12.optPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.12.lnbMeasVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.12.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.tmpFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.status	TEXT	R/O	StringRange R/O
psu.1.enable	CHOICE	R/O	EnumRange R/O (PRESENT NOT PRESENT)
psu.1.name	TEXT	R/O	StringRange R/O
psu.1.voltage1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu.1.voltage2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu.1.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
psu.2.enable	CHOICE	R/O	EnumRange R/O (PRESENT NOT PRESENT)
psu.2.name	TEXT	R/O	StringRange R/O
psu.2.voltage1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu.2.voltage2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu.2.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 6)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 7)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 8)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 9)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 11)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 12)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Main 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Main 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Standby)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Remote/Override)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Main 1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Main 2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Standby)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Remote/Override)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Main 1)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Main 2)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Standby)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Remote/Override)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 11 Main 1)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 11 Main 2)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 11 Standby)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 11 Remote/Override)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 1 Card Comms)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Card Comms)

name	*type*	*flags*	*range*
faults.33	ALARM	R/O	AlarmFlagRange R/O (Slot 3 Card Comms)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Slot 4 Card Comms)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Card Comms)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Slot 6 Card Comms)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Slot 7 Card Comms)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Card Comms)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Slot 9 Card Comms)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Slot 10 Card Comms)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Slot 11 Card Comms)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Slot 12 Card Comms)

9.254 Frequency-Loop

The logical device Frequency-Loop is used to configure the receive frequency of receivers or demodulators for a test loop. Normally the receiver will be configured in terms of L-Band loop.

The Frequency-Loop device takes the configured LO frequency of transmitter into account to calculate the the L-Band frequency which has to be set at the receiver. If the LO is zero it will simple use the RF frequency directly.

The device monitors transmit frequency, transmit LO and LO polarity of the transmitter and calculates the output l-band frequency. If the user presses the SET LOOP button, the device saves the current LO frequencies of the receiver internally, sets the LO frequencies at the receiver to 0 and the receive frequency to the calculated l-band loop frequency. After that it restores the LO frequency settings at the receiver.

Device window pages

The following table shows which device window pages are available with this individual device

type.

- --- RX and TX frequencies
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
txFrequencyId	ID of parameter to read transmit frequency from (e.g. ENC.mod.frequency)
rxFrequencyId	ID of parameter to set the receive frequency (e.g IRD.frequency)
loTxFreqId	ID of parameter to read the transmitter LO frequency
loRxFreqHighId	ID of parameter to read and set the receiver high-band LO frequency
loRxFreqLowId	ID of parameter to read and set the receiver low-band LO frequency
txFrequencyId	Polarization of the LO frequency (Normal or Inverted)

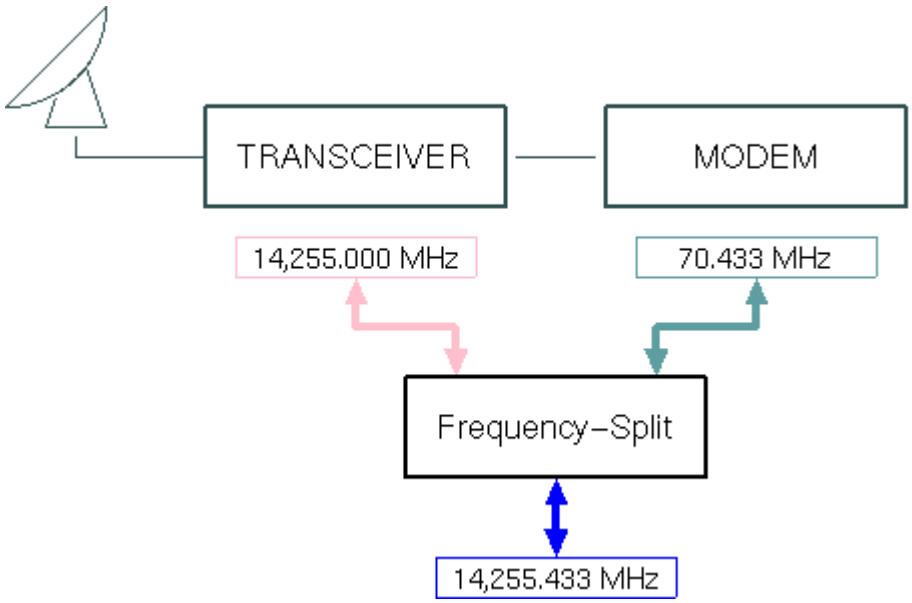
Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.rxFrequencyId	TEXT	SAVE SETUP	StringRange
config.loTxFreqId	TEXT	SAVE SETUP	StringRange
config.loRxFreqHighId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.loRxFreqLowId	TEXT	SAVE SETUP	StringRange
config.loRxBandEdge	TEXT	SAVE SETUP	StringRange
config.loPol	CHOICE	SAVE SETUP	EnumRange (NORMAL INVERT)
config.irdMode	CHOICE	SAVE SETUP	EnumRange (RF L-BAND)
take	TEXT		StringRange
txFrequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
rxFrequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
lBandFrequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
loTxFreq	FLOAT		DoubleRange (0.0000 .. 0.0000)
loRxFreqHigh	FLOAT		DoubleRange (0.0000 .. 0.0000)
loRxFreqLow	FLOAT		DoubleRange (0.0000 .. 0.0000)
loRxBandEdge	FLOAT		DoubleRange (0.0000 .. 0.0000)
internal.low	FLOAT		DoubleRange (0.0000 .. 0.0000)
internal.high	FLOAT		DoubleRange (0.0000 .. 0.0000)

9.255 Frequency-Split

The Frequency-Split logical device splits up a frequency value in a coarse part (1 MHz resolution) and a fine part. The coarse part is set at the RF device while the fine part is set as an offset to the nominal IF frequency at a device at the IF stage. A typical application for the Frequency-Split device is to set the transmit/receive frequencies with a transceiver that only is capable to tune it's frequencies in steps of 1 MHz. The figure below shows the usage of the Frequency-Split device in an up-link application, however it may be used for down-link frequencies as well.



The Frequency-Split device processes the frequency setting messages in both directions. If you change the frequencies directly at the devices, the Frequency-Split recognizes this and updates the common frequency setting accordingly.

When setting the coarse part of the frequency, the Frequency-Split rounds to the nearest full MHz. Hence the offset applied to the IF frequency setting will be +/- 500 kHz.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control page for the device. The only operational parameter shown on this page is the common frequency value.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Frequency-Split device with the devices that will perform the frequency setting.

parameter	***description***
fineFreqId	Enter the message identifier here of the parameter which shall receive the fine part frequency setting, e.g. "MODEM-1.tx.frequency". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.

parameter	*description*
coarseFreqId	Enter the message identifier here of the parameter which shall receive the coarse part frequency setting, e.g. "TRANSCEIVER.tx.frequency". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
nominalFreq	The nominal IF frequency to be set at the "fineFreqId" parameter must be specified here.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.fineFreqId	TEXT	SAVE SETUP	StringRange
config.coarseFreqId	TEXT	SAVE SETUP	StringRange
config.nominalFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 30000.000)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)

9.256 GDSatcom-ACU930

This page describes the device driver and the device window for the GDSatcom antenna controller 930. The driver supports the steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the preset position memories of the ACU, lets you investigate the contents of the preset memories, store edited presets and execute presets.
- --- The third parameter page displays the tracking parameters and allows to start / stop resume OPTrack and steptrack tracking. The beacon receiver related parameters also are located at this page.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Controller.

parameter	***description***
hasPolAxis	En- or Disables the polarization axis
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The GDSatcom ACU 930 is controlled through a TCP connection to one of the ports 5001 or 5002 of the device. Each port only accepts one connection at a time. The communication has to be configured as to a serial terminal server port, specifying the device address and port number in the device configuration.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.versionAcu	TEXT	R/O	StringRange R/O
info.versionIo	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.hasPolAxis	CHOICE	R/O	EnumRange R/O (YES NO)
info.polOption	CHOICE	R/O	EnumRange R/O (NO CP LP CP/LP)
info.hasTableTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasNoradTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasIntelsatTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.diameter	FLOAT		DoubleRange (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
config.hasPolAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.19	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.20	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.21	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.22	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.23	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.24	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.25	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.26	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.27	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.28	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.29	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.30	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.31	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.32	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.33	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.34	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.35	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.36	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.37	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.38	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.39	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.40	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.41	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.42	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.43	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.44	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.45	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.46	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.47	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.48	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.49	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.50	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.51	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.52	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.53	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.54	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.55	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.56	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.57	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.58	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.59	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.60	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.61	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.62	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.63	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.64	HEX	R/O	HexRange R/O (0 .. 0)
internal.prf1	TEXT	R/O	StringRange R/O
internal.prf2	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.name9	TEXT	R/O	StringRange R/O
internal.name10	TEXT	R/O	StringRange R/O
internal.name11	TEXT	R/O	StringRange R/O
internal.name12	TEXT	R/O	StringRange R/O
internal.name13	TEXT	R/O	StringRange R/O
internal.name14	TEXT	R/O	StringRange R/O
internal.name15	TEXT	R/O	StringRange R/O
internal.name16	TEXT	R/O	StringRange R/O
internal.name17	TEXT	R/O	StringRange R/O
internal.name18	TEXT	R/O	StringRange R/O
internal.name19	TEXT	R/O	StringRange R/O
internal.name20	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.azHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.pol	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.submode1	TEXT	R/O	StringRange R/O
state.submode2	TEXT	R/O	StringRange R/O
state.submode3	TEXT	R/O	StringRange R/O
state.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.target	TEXT	R/O	StringRange R/O
state.polPlane	TEXT	R/O	StringRange R/O
prespos.no	INTEGER		IntegerRange (1 .. 20)
prespos.byname	CHOICE		EnumRange (?)
prespos.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prespos.az	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.el	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.pl	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.name	TEXT		StringRange
prespos.execute	TEXT	NOPRESET	StringRange
prespos.store	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
steptrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
steptrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
steptrack.size	FLOAT		DoubleRange (5.0 .. 20.0)
steptrack.sct	FLOAT		DoubleRange (0.0 .. 240.0)
steptrack.snt	INTEGER		IntegerRange (1 .. 4)
steptrack.pattern	CHOICE		EnumRange (NO SCAN BOX SCAN)
steptrack.execute	TEXT	NOPRESET	StringRange
steptrack.store	TEXT	NOPRESET	StringRange
steptrack.cancel	TEXT	NOPRESET	StringRange
steptrack.tracknow	TEXT	NOPRESET	StringRange
rf.no	INTEGER		IntegerRange (1 .. 20)
rf.trackMode	CHOICE		EnumRange (DIRECT PRESET INTELSAT NORAD)
rf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
rf.store	TEXT	NOPRESET	StringRange
rf.frequency	FLOAT		DoubleRange (100.000 .. 100000.000)
rf.slope	FLOAT		DoubleRange (-100.00 .. 100.00)
rf.offset	FLOAT		DoubleRange (-100.00 .. 100.00)
rf.sigThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
rf.acqThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
rf.polAbSwitch	CHOICE		EnumRange (0 1)
rf.bandwidth	CHOICE		EnumRange (1 kHz 4 kHz)
rf.beaconChoice	CHOICE		EnumRange (1 2 3 4)
rf.polMode	CHOICE		EnumRange (STOP POS DESIGNATE AUTO)

name	*type*	*flags*	*range*
rf.polPosition	FLOAT		DoubleRange (0.00 .. 0.00)
rf.polPlane	CHOICE		EnumRange (H V RHCP LHCP)
rf.polSwitch	CHOICE		EnumRange (CP LP)
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ Drive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (El Drive)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Interlock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Resolver)
faults.06	ALARM	R/O	AlarmFlagRange R/O (EL Resolver)
faults.07	ALARM	R/O	AlarmFlagRange R/O (POL Resolver)
faults.08	ALARM	R/O	AlarmFlagRange R/O (CP/LP Switch)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ/EL OE Mismatch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (FPGA Type)
faults.11	ALARM	R/O	AlarmFlagRange R/O (FPGA Config)
faults.12	ALARM	R/O	AlarmFlagRange R/O (FPGA PS)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Loss Sync)
faults.14	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Inactive)
faults.15	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Error1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Error2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Loss Sync)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Inactive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Error1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Error2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (AZ Immobile)
faults.22	ALARM	R/O	AlarmFlagRange R/O (EL Immobile)
faults.23	ALARM	R/O	AlarmFlagRange R/O (AZ Reverse Dir)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EL Reverse Dir)
faults.25	ALARM	R/O	AlarmFlagRange R/O (AZ Unexpected Motion)
faults.26	ALARM	R/O	AlarmFlagRange R/O (EL Unexpected Motion)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AZ CW Prelimit)
faults.28	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Prelimit)
faults.29	ALARM	R/O	AlarmFlagRange R/O (EL Up Prelimit)
faults.30	ALARM	R/O	AlarmFlagRange R/O (EL Dwn Prelimit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (AZ CW SW Limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (AZ CCW SW Limit)
faults.33	ALARM	R/O	AlarmFlagRange R/O (EL Up SW Limit)
faults.34	ALARM	R/O	AlarmFlagRange R/O (EL Dwn SW Limit)
faults.35	ALARM	R/O	AlarmFlagRange R/O (POL CW Prelimit)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (POL CCW Prelimit)
faults.37	ALARM	R/O	AlarmFlagRange R/O (POL CW SW Limit)
faults.38	ALARM	R/O	AlarmFlagRange R/O (POL CCW SW Limit)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Simulation On)
faults.40	ALARM	R/O	AlarmFlagRange R/O (ACU-PDU Link)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Websock)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Receiver Broadcast)
faults.43	ALARM	R/O	AlarmFlagRange R/O (DSP)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Load Site Location)
faults.45	ALARM	R/O	AlarmFlagRange R/O (ACU Software)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Execute Motion Disable)
faults.47	ALARM	R/O	AlarmFlagRange R/O (NTP Task)
faults.48	ALARM	R/O	AlarmFlagRange R/O (CLI Task)
faults.49	ALARM	R/O	AlarmFlagRange R/O (MODE Task)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Database Task)
faults.51	ALARM	R/O	AlarmFlagRange R/O (CP Task)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Keypad Task)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Status Task)
faults.55	ALARM	R/O	AlarmFlagRange R/O (MEMTRK Task)
faults.56	ALARM	R/O	AlarmFlagRange R/O (SIG PROCESS)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Tracking Task)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Console CLI)

name	*type*	*flags*	*range*
faults.59	ALARM	R/O	AlarmFlagRange R/O (Simulation Task)
faults.60	ALARM	R/O	AlarmFlagRange R/O (UPC Task)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Websock Task)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Execute Stop Mode)
faults.66	ALARM	R/O	AlarmFlagRange R/O (DBKUP Task)
faults.67	ALARM	R/O	AlarmFlagRange R/O (UPC Link Down)
faults.68	ALARM	R/O	AlarmFlagRange R/O (UPC)
faults.69	ALARM	R/O	AlarmFlagRange R/O (ACU-GPS Link)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Low signal level)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Summary fault)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.73	ALARM	R/O	AlarmFlagRange R/O (RCV 1 Link)
faults.74	ALARM	R/O	AlarmFlagRange R/O (RCV 2 Link)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Invalid DB)
faults.76	ALARM	R/O	AlarmFlagRange R/O (RCV 1 Fault)
faults.77	ALARM	R/O	AlarmFlagRange R/O (RCV 2 Fault)

9.257 GDSatcom-ACU940

This page describes the device driver and the device window for the GDSatcom antenna controller 940. The driver supports the OPTrack and steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the preset position memories of the ACU, lets you investigate the contents of the preset memories, store edited presets and execute presets.
- --- The third parameter page displays the tracking parameters and allows to start / stop resume OPTrack and steptrack tracking. The beacon receiver related parameters also are located at this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Controller.

parameter	***description***
hasPolAxis	En- or Disables the polarization axis
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The GDSatcom ACU 940 is controlled through a TCP connection to one of the ports 5001 or 5002 of the device. Each port only accepts one connection at a time. The communication has to be configured as to a serial terminal server port, specifying the device address and port number in the device configuration.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.versionAcu	TEXT	R/O	StringRange R/O
info.versionIo	TEXT	R/O	StringRange R/O
info.hasPolAxis	CHOICE	R/O	EnumRange R/O (YES NO)
info.polOption	CHOICE	R/O	EnumRange R/O (NO CP LP CP/LP)
info.hasTableTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasNoradTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.hasIntelsatTrk	CHOICE	R/O	EnumRange R/O (YES NO)
info.diameter	FLOAT		DoubleRange (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
config.hasPolAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.azJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.19	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.20	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.21	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.22	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.23	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.24	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.25	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.26	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.27	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.28	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.29	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.30	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.31	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.32	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.33	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.34	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.35	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.36	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.37	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.38	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.39	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.40	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.41	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.42	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.43	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.44	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.45	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.46	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.47	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.48	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.49	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.50	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.51	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.52	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.53	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.54	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.55	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.56	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.57	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.58	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.59	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.60	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.61	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.62	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.63	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.64	HEX	R/O	HexRange R/O (0 .. 0)
internal.prf1	TEXT	R/O	StringRange R/O
internal.prf2	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.name9	TEXT	R/O	StringRange R/O
internal.name10	TEXT	R/O	StringRange R/O
internal.name11	TEXT	R/O	StringRange R/O
internal.name12	TEXT	R/O	StringRange R/O
internal.name13	TEXT	R/O	StringRange R/O
internal.name14	TEXT	R/O	StringRange R/O
internal.name15	TEXT	R/O	StringRange R/O
internal.name16	TEXT	R/O	StringRange R/O
internal.name17	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.name18	TEXT	R/O	StringRange R/O
internal.name19	TEXT	R/O	StringRange R/O
internal.name20	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.azHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.pol	INTEGER	R/O	IntegerRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.submode1	TEXT	R/O	StringRange R/O
state.submode2	TEXT	R/O	StringRange R/O
state.submode3	TEXT	R/O	StringRange R/O
state.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.target	TEXT	R/O	StringRange R/O
state.polPlane	TEXT	R/O	StringRange R/O
prespos.no	INTEGER		IntegerRange (1 .. 20)
prespos.byname	CHOICE		EnumRange (?)
prespos.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prespos.az	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.el	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.pl	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.name	TEXT		StringRange
prespos.execute	TEXT	NOPRESET	StringRange
prespos.store	TEXT	NOPRESET	StringRange
prespos.polMode	CHOICE		EnumRange (STOP POS DESIG AUTO)
prespos.polPosition	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.polPlane	CHOICE		EnumRange (H V RHCP LHCP)
prespos.polSwitch	CHOICE		EnumRange (CP LP)
optrack.no	INTEGER		IntegerRange (1 .. 10)
optrack.byname	CHOICE		EnumRange (?)
optrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
optrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
optrack.sct	CHOICE		EnumRange (10 min 20 min 30 min)
optrack.pattern	CHOICE		EnumRange (NO SCAN BOX SCAN)
optrack.name	TEXT		StringRange
optrack.execute	TEXT	NOPRESET	StringRange
optrack.store	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
steptrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
steptrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
steptrack.size	FLOAT		DoubleRange (5.0 .. 20.0)
steptrack.sct	FLOAT		DoubleRange (0.0 .. 240.0)
steptrack.snt	INTEGER		IntegerRange (1 .. 4)
steptrack.pattern	CHOICE		EnumRange (NO SCAN BOX SCAN)
steptrack.execute	TEXT	NOPRESET	StringRange
steptrack.store	TEXT	NOPRESET	StringRange
steptrack.cancel	TEXT	NOPRESET	StringRange
steptrack.tracknow	TEXT	NOPRESET	StringRange
rf.no	INTEGER		IntegerRange (1 .. 20)
rf.trackMode	CHOICE		EnumRange (DIRECT PRESET INTELSAT OPTRACK NORAD)
rf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
rf.store	TEXT	NOPRESET	StringRange
rf.frequency	FLOAT		DoubleRange (100.000 .. 100000.000)
rf.slope	FLOAT		DoubleRange (-100.00 .. 100.00)
rf.offset	FLOAT		DoubleRange (-100.00 .. 100.00)
rf.sigThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
rf.acqThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
rf.polAbSwitch	CHOICE		EnumRange (0 1)
rf.bandwidth	CHOICE		EnumRange (1 kHz 4 kHz 250 kHz 280 kHz)
rf.beaconChoice	CHOICE		EnumRange (N/A 1 2 3 4)
rf.sweepWidth	FLOAT		DoubleRange (40.0 .. 300.0)
rf.polMode	CHOICE		EnumRange (STOP POS DESIG AUTO)

name	*type*	*flags*	*range*
rf.polPosition	FLOAT		DoubleRange (0.00 .. 0.00)
rf.polPlane	CHOICE		EnumRange (H V RHCP LHCP)
rf.polSwitch	CHOICE		EnumRange (CP LP)
geo.latitude	FLOAT		DoubleRange (-90.0000 .. 90.0000)
geo.longitude	FLOAT		DoubleRange (0.000 .. 360.000)
geo.azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
geo.elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
geo.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
geo.execute	TEXT	NOPRESET	StringRange
manpos	TEXT	NOPRESET	StringRange
step	TEXT	NOPRESET	StringRange
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ Drive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (El Drive)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Interlock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Resolver)
faults.06	ALARM	R/O	AlarmFlagRange R/O (EL Resolver)
faults.07	ALARM	R/O	AlarmFlagRange R/O (POL Resolver)
faults.08	ALARM	R/O	AlarmFlagRange R/O (CP/LP Switch)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ/EL OE Mismatch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (FPGA Type)
faults.11	ALARM	R/O	AlarmFlagRange R/O (FPGA Config)
faults.12	ALARM	R/O	AlarmFlagRange R/O (FPGA PS)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Loss Sync)
faults.14	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Inactive)
faults.15	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Error1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder Error2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Loss Sync)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Inactive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Error1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Error2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (AZ Immobile)
faults.22	ALARM	R/O	AlarmFlagRange R/O (EL Immobile)
faults.23	ALARM	R/O	AlarmFlagRange R/O (AZ Reverse Dir)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EL Reverse Dir)
faults.25	ALARM	R/O	AlarmFlagRange R/O (AZ Unexpected Motion)
faults.26	ALARM	R/O	AlarmFlagRange R/O (EL Unexpected Motion)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AZ CW Prelimit)
faults.28	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Prelimit)
faults.29	ALARM	R/O	AlarmFlagRange R/O (EL Up Prelimit)
faults.30	ALARM	R/O	AlarmFlagRange R/O (EL Dwn Prelimit)

name	*type*	*flags*	*range*
faults.31	ALARM	R/O	AlarmFlagRange R/O (AZ CW SW Limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (AZ CCW SW Limit)
faults.33	ALARM	R/O	AlarmFlagRange R/O (EL Up SW Limit)
faults.34	ALARM	R/O	AlarmFlagRange R/O (EL Dwn SW Limit)
faults.35	ALARM	R/O	AlarmFlagRange R/O (POL CW Prelimit)
faults.36	ALARM	R/O	AlarmFlagRange R/O (POL CCW Prelimit)
faults.37	ALARM	R/O	AlarmFlagRange R/O (POL CW SW Limit)
faults.38	ALARM	R/O	AlarmFlagRange R/O (POL CCW SW Limit)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Simulation On)
faults.40	ALARM	R/O	AlarmFlagRange R/O (ACU-PDU Link)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Websock)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Receiver Broadcast)
faults.43	ALARM	R/O	AlarmFlagRange R/O (DSP)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Load Site Location)
faults.45	ALARM	R/O	AlarmFlagRange R/O (ACU Software)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Execute Motion Disable)
faults.47	ALARM	R/O	AlarmFlagRange R/O (NTP Task)
faults.48	ALARM	R/O	AlarmFlagRange R/O (CLI Task)
faults.49	ALARM	R/O	AlarmFlagRange R/O (MODE Task)

name	*type*	*flags*	*range*
faults.50	ALARM	R/O	AlarmFlagRange R/O (Database Task)
faults.51	ALARM	R/O	AlarmFlagRange R/O (CP Task)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Keypad Task)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Status Task)
faults.54	ALARM	R/O	AlarmFlagRange R/O (OPTRACK Task)
faults.55	ALARM	R/O	AlarmFlagRange R/O (MEMTRK Task)
faults.56	ALARM	R/O	AlarmFlagRange R/O (SIG PROCESS)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Tracking Task)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Console CLI)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Simulation Task)
faults.60	ALARM	R/O	AlarmFlagRange R/O (UPC Task)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Websock Task)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Execute Stop Mode)
faults.63	ALARM	R/O	AlarmFlagRange R/O (OPTRACK Table)
faults.64	ALARM	R/O	AlarmFlagRange R/O (OPTRACK Data)
faults.65	ALARM	R/O	AlarmFlagRange R/O (OPTRACK Pol)
faults.66	ALARM	R/O	AlarmFlagRange R/O (DBKUP Task)
faults.67	ALARM	R/O	AlarmFlagRange R/O (UPC Link Down)
faults.68	ALARM	R/O	AlarmFlagRange R/O (UPC)
faults.69	ALARM	R/O	AlarmFlagRange R/O (ACU-GPS Link)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Low signal level)

name	*type*	*flags*	*range*
faults.71	ALARM	R/O	AlarmFlagRange R/O (Summary fault)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.73	ALARM	R/O	AlarmFlagRange R/O (RCV 1 Link)
faults.74	ALARM	R/O	AlarmFlagRange R/O (RCV 2 Link)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Invalid DB)
faults.76	ALARM	R/O	AlarmFlagRange R/O (RCV 1 Fault)
faults.77	ALARM	R/O	AlarmFlagRange R/O (RCV 2 Fault)

9.258 GDSatcom-ACU950

This page describes the device driver and the device window for the GDSatcom antenna controller ACU950. The driver supports the OPTrack and steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the preset position memories of the ACU, lets you investigate the contents of the preset memories, store edited presets and execute presets.
- --- The third parameter page displays the tracking parameters and allows to start / stop resume OPTrack and steptrack tracking. The beacon receiver related parameters also are located at this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

There are no specific configuration parameters for this device.

Remarks

The GDSatcom ACU 950 is controlled through a TCP connection to one of the ports 5001 or 5002 of the device. Each port only accepts one connetion at a time. The communication has to be configured as to a serial terminal server port, specifying the device address and port number in the device configuration.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.hasPolAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
internal.destUpdated	TEXT	R/O	StringRange R/O
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.19	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.20	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.21	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.22	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.23	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.24	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.25	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.26	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.27	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.28	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.29	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.30	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.31	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.32	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.33	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.34	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.35	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.36	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.37	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.38	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.39	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.40	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.41	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.42	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.43	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.44	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.45	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.46	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.47	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.48	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.49	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.50	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.51	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.52	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.53	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.54	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.55	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.56	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.57	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.58	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.59	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.60	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.61	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.62	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.63	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.64	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.65	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.66	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.67	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.68	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.69	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.70	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.71	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.72	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.73	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.74	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.75	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.76	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.77	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.78	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.79	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.80	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.89	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.92	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.96	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.98	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.100	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.102	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.103	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.104	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.105	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.108	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.109	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.113	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.116	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.117	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.118	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.121	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.125	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.126	HEX	R/O	HexRange R/O (0 .. 0)
internal.prf1	TEXT	R/O	StringRange R/O
internal.prf2	TEXT	R/O	StringRange R/O
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.submode1	TEXT	R/O	StringRange R/O
state.submode2	TEXT	R/O	StringRange R/O
state.submode3	TEXT	R/O	StringRange R/O
state.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
prespos.no	INTEGER		IntegerRange (1 .. 40)
prespos.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prespos.az	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.el	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.pl	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.name	TEXT		StringRange
prespos.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
prespos.cw	CHOICE		EnumRange (CLOCKWISE COUNTERCLOCKWISE SHORTEST)
prespos.execute	TEXT		StringRange
prespos.store	TEXT		StringRange
optrack.no	INTEGER		IntegerRange (1 .. 3)
optrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
optrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
optrack.sct	CHOICE		EnumRange (10 min 20 min 30 min)
optrack.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
optrack.name	TEXT		StringRange
optrack.execute	TEXT		StringRange
optrack.store	TEXT		StringRange
steptrack.no	INTEGER		IntegerRange (1 .. 10)
steptrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
steptrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
steptrack.snt	INTEGER		IntegerRange (1 .. 4)
steptrack.size	FLOAT		DoubleRange (5.0 .. 20.0)

name	*type*	*flags*	*range*
steptrack.data	CHOICE		EnumRange (OFF ON)
steptrack.sct	FLOAT		DoubleRange (0.0 .. 240.0)
steptrack.gain	FLOAT		DoubleRange (0.00 .. 1.00)
steptrack.scan	CHOICE		EnumRange (OFF ON)
steptrack.scanNum	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 AUTO)
steptrack.park	FLOAT		DoubleRange (3.0 .. 10.0)
steptrack.limit	FLOAT		DoubleRange (0.0 .. 10.0)
steptrack.name	TEXT		StringRange
steptrack.execute	TEXT		StringRange
steptrack.store	TEXT		StringRange
steptrack.cancel	TEXT		StringRange
steptrack.tracknow	TEXT		StringRange
cfg.no	INTEGER		IntegerRange (1 .. 20)
cfg.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
cfg.acqThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.sigThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.name	TEXT		StringRange
cfg.execute	TEXT		StringRange
cfg.store	TEXT		StringRange
rf.no	INTEGER		IntegerRange (1 .. 20)
rf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
rf.frequency	FLOAT		DoubleRange (100.000 .. 100000.000)
rf.bandwidth	CHOICE		EnumRange (62.5 Hz 125 Hz 250 Hz 500 Hz 1 kHz 4 kHz 250kHz)
rf.slope	FLOAT		DoubleRange (0.1000 .. 100.0000)
rf.offset	FLOAT		DoubleRange (-100.0000 .. 100.0000)

name	*type*	*flags*	*range*
rf.activeSearch	CHOICE		EnumRange (OFF ON)
rf.feedOffsetEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.feedOffsetXEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.path	CHOICE		EnumRange (1 2)
rf.sweepWidth	FLOAT		DoubleRange (40.0 .. 145.0)
rf.name	TEXT		StringRange
rf.bdcFrequency	FLOAT		DoubleRange (0.000 .. 100000.000)
rf.execute	TEXT		StringRange
rf.store	TEXT		StringRange
profile	TEXT	R/O	StringRange R/O
prf.select	CHOICE		EnumRange (0 - None)
prf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prf.no	INTEGER		IntegerRange (1 .. 20)
prf.name	TEXT		StringRange
prf.pointMode	CHOICE		EnumRange (STOP POS DESIGNATE PRESET POSITION GEO PRESET INTELSAT TRACK GEO DESIGNATE MANUAL POSITION STOW OPTRACK STAR TRACK TABLE TRACK NORAD TRACK MANUAL RATE SUN TRACK MOON TRACK POL AUTO)
prf.pointData	INTEGER		IntegerRange (0 .. 0)
prf.acquireMode	CHOICE		EnumRange (NONE BOX SCAN GEO SCAN)
prf.acquireData	INTEGER		IntegerRange (0 .. 11)
prf.trackMode	CHOICE		EnumRange (unknown NONE STEPTRACK)
prf.trackData	INTEGER		IntegerRange (0 .. 11)
prf.rfData	INTEGER		IntegerRange (0 .. 20)
prf.configData	INTEGER		IntegerRange (0 .. 20)

name	*type*	*flags*	*range*
prf.polMode	CHOICE		EnumRange (unknown STOP PRE MANRATE MAN POS OPTRACK INTLSAT TABTRAK AUTO)
prf.polData	INTEGER		IntegerRange (0 .. 10)
prf.userInfo	TEXT		StringRange
prf.save	TEXT		StringRange
prf.activate	TEXT		StringRange
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ CW Limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ CW Prelimit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Prelimit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Brake)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Control)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Handcrank)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ Motor 3-Phase OFF)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL Up Limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL Down Limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL Up Prelimit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (EL Down Prelimit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (EL Summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL Brake)
faults.15	ALARM	R/O	AlarmFlagRange R/O (EL Motor Control)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (EL Motor Handcrank)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL Motor 3-Phase OFF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (POL CW Limit)
faults.19	ALARM	R/O	AlarmFlagRange R/O (POL CCW Limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (POL Brake)
faults.21	ALARM	R/O	AlarmFlagRange R/O (POL Motor Overload)
faults.22	ALARM	R/O	AlarmFlagRange R/O (POL2 CW Limit)
faults.23	ALARM	R/O	AlarmFlagRange R/O (POL2 CCW Limit)
faults.24	ALARM	R/O	AlarmFlagRange R/O (POL2 Motor Overload)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Interlock Logic)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Safety Circuit)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AZ Resolver)
faults.28	ALARM	R/O	AlarmFlagRange R/O (EL Resolver)
faults.29	ALARM	R/O	AlarmFlagRange R/O (ACU Stop Switch)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop 1)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop 2)
faults.32	ALARM	R/O	AlarmFlagRange R/O (PDU Emergency Stop)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Motor Overtemp)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Fan 2)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (ACB DSER Hot)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Simulation ON)
faults.38	ALARM	R/O	AlarmFlagRange R/O (AZ Oscillation)
faults.39	ALARM	R/O	AlarmFlagRange R/O (EL Oscillation)
faults.40	ALARM	R/O	AlarmFlagRange R/O (POL Oscillation)
faults.41	ALARM	R/O	AlarmFlagRange R/O (POL2 Oscillation)
faults.42	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder)
faults.43	ALARM	R/O	AlarmFlagRange R/O (EL Encoder)
faults.44	ALARM	R/O	AlarmFlagRange R/O (AZ Tach)
faults.45	ALARM	R/O	AlarmFlagRange R/O (EL Tach)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Low Signal)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Unacquirable signal)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Travel Limit AZ CW)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Travel Limit AZ CCW)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Travel Limit EL Up)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Travel Limit EL Down)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Travel Limit PL Up)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Travel Limit PL Down)
faults.55	ALARM	R/O	AlarmFlagRange R/O (AZ Runaway)
faults.56	ALARM	R/O	AlarmFlagRange R/O (EL Runaway)

name	*type*	*flags*	*range*
faults.57	ALARM	R/O	AlarmFlagRange R/O (Receiver Datalink)
faults.58	ALARM	R/O	AlarmFlagRange R/O (AZ Varac Drv Link Down)
faults.59	ALARM	R/O	AlarmFlagRange R/O (EL Varac Drv Link Down)
faults.60	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CW)
faults.61	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CCW)
faults.62	ALARM	R/O	AlarmFlagRange R/O (SW Limit EL Up)
faults.63	ALARM	R/O	AlarmFlagRange R/O (SW Limit EL Down)
faults.64	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL CW)
faults.65	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL CCW)
faults.66	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL2 CW)
faults.67	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL2 CCW)
faults.68	ALARM	R/O	AlarmFlagRange R/O (ACB Link Down)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Bad Track Point)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Load Site Location)
faults.71	ALARM	R/O	AlarmFlagRange R/O (AZ Varac Drv Setup)
faults.72	ALARM	R/O	AlarmFlagRange R/O (EL Varac Drv Setup)
faults.73	ALARM	R/O	AlarmFlagRange R/O (ACB SW)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Execute Stop Mode)

name	*type*	*flags*	*range*
faults.83	ALARM	R/O	AlarmFlagRange R/O (Execute Stop Mode SW)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Disable AZ Key)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Disable EL Key)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Disable PL Key)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Optrack Table)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Optrack AZEL Data)
faults.80	ALARM	R/O	AlarmFlagRange R/O (DB Backup Task)
faults.81	ALARM	R/O	AlarmFlagRange R/O (ACB1 Heartbeat Lost)
faults.82	ALARM	R/O	AlarmFlagRange R/O (ACB2 Heartbeat Lost)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Real Time Clock)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Real Time Clock Stuck)
faults.86	ALARM	R/O	AlarmFlagRange R/O (XCO Time Error)
faults.87	ALARM	R/O	AlarmFlagRange R/O (NTP Time)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Unexpexted Motion AZ)
faults.89	ALARM	R/O	AlarmFlagRange R/O (Unexpexted Motion EL)
faults.90	ALARM	R/O	AlarmFlagRange R/O (RCVR)
faults.91	ALARM	R/O	AlarmFlagRange R/O (Threshold AZ+)
faults.92	ALARM	R/O	AlarmFlagRange R/O (Threshold AZ-)
faults.93	ALARM	R/O	AlarmFlagRange R/O (Threshold EL+)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Threshold EL-)
faults.95	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal AZ)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal EL)

name	*type*	*flags*	*range*
faults.97	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal PL)

9.259 GDSatcom-ACU970

This page describes the device driver and the device window for the GDSatcom antenna controller ACU970. The driver supports the monopulse, OPTrack and steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the preset position memories of the ACU, lets you investigate the contents of the preset memories, store edited presets and execute presets.
- --- The third parameter page displays the tracking parameters and allows to start / stop resume OPTrack and steptrack tracking. The beacon receiver related parameters also are located at this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

There are no specific configuration parameters for this device.

Remarks

The GDSatcom ACU 970 is controlled through a TCP connection to one of the ports 5001 or 5002 of the device. Each port only accepts one connection at a time. The communication has to be configured as to a serial terminal server port, specifying the device address and port number in the device configuration.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.hasPolAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
internal.destUpdated	TEXT	R/O	StringRange R/O
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.19	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.20	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.21	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.22	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.23	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.24	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.25	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.26	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.27	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.28	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.29	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.30	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.31	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.32	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.33	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.34	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.35	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.36	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.37	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.38	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.39	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.40	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.41	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.42	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.43	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.44	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.45	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.46	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.47	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.48	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.49	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.50	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.51	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.52	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.53	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.54	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.55	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.56	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.57	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.58	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.59	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.60	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.61	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.62	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.63	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.64	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.65	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.66	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.67	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.68	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.69	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.70	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.71	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.72	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.73	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.74	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.75	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.76	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.77	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.78	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.79	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.80	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.89	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.92	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.93	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.98	HEX	R/O	HexRange R/O (0 .. 0)
internal.neg	HEX	R/O	HexRange R/O (0 .. 0)
internal.prf1	TEXT	R/O	StringRange R/O
internal.prf2	TEXT	R/O	StringRange R/O
internal.trstat.1	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.2	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.3	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.7	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.8	HEX	R/O	HexRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.submode1	TEXT	R/O	StringRange R/O
state.submode2	TEXT	R/O	StringRange R/O
state.submode3	TEXT	R/O	StringRange R/O
state.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.elSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.polSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.polPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.elError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.polError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.scanMode	TEXT	R/O	StringRange R/O
state.phaseMode	TEXT	R/O	StringRange R/O
state.rfSwitch	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.sumRhcp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sumLhcp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sumChannel	CHOICE	R/O	EnumRange R/O (UNKNOWN RHCP LHCP)
prespos.no	INTEGER		IntegerRange (1 .. 40)
prespos.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prespos.az	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.el	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.pl	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.name	TEXT		StringRange
prespos.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
prespos.cw	CHOICE		EnumRange (CLOCKWISE COUNTERCLOCKWISE SHORTEST)
prespos.execute	TEXT		StringRange
prespos.store	TEXT		StringRange
optrack.no	INTEGER		IntegerRange (1 .. 3)
optrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
optrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
optrack.sct	CHOICE		EnumRange (10 min 20 min 30 min)
optrack.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
optrack.tracktype	CHOICE		EnumRange (STEPTRACK MONOPULSE)
optrack.name	TEXT		StringRange
optrack.execute	TEXT		StringRange
optrack.store	TEXT		StringRange
steptrack.no	INTEGER		IntegerRange (1 .. 10)

name	*type*	*flags*	*range*
steptrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
steptrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
steptrack.snt	INTEGER		IntegerRange (1 .. 4)
steptrack.size	FLOAT		DoubleRange (5.0 .. 20.0)
steptrack.data	CHOICE		EnumRange (OFF ON)
steptrack.sct	FLOAT		DoubleRange (0.0 .. 240.0)
steptrack.gain	FLOAT		DoubleRange (0.00 .. 1.00)
steptrack.scan	CHOICE		EnumRange (OFF ON)
steptrack.scanNum	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 AUTO)
steptrack.park	FLOAT		DoubleRange (3.0 .. 10.0)
steptrack.limit	FLOAT		DoubleRange (0.0 .. 10.0)
steptrack.name	TEXT		StringRange
steptrack.execute	TEXT		StringRange
steptrack.store	TEXT		StringRange
steptrack.cancel	TEXT		StringRange
steptrack.tracknow	TEXT		StringRange
cfg.no	INTEGER		IntegerRange (1 .. 20)
cfg.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
cfg.acqThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.sigThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.name	TEXT		StringRange
cfg.execute	TEXT		StringRange
cfg.store	TEXT		StringRange
rf.no	INTEGER		IntegerRange (1 .. 20)
rf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
rf.frequency	FLOAT		DoubleRange (100.000 .. 100000.000)

name	*type*	*flags*	*range*
rf.bandwidth	CHOICE		EnumRange (62.5 Hz 125 Hz 250 Hz 500 Hz 1 kHz 4 kHz 250kHz)
rf.slope	FLOAT		DoubleRange (0.1000 .. 100.0000)
rf.offset	FLOAT		DoubleRange (-100.0000 .. 100.0000)
rf.activeSearch	CHOICE		EnumRange (OFF ON)
rf.feedOffsetEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.feedOffsetXEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.path	CHOICE		EnumRange (1 2)
rf.sweepWidth	FLOAT		DoubleRange (40.0 .. 145.0)
rf.name	TEXT		StringRange
rf.bdcFrequency	FLOAT		DoubleRange (0.000 .. 100000.000)
rf.execute	TEXT		StringRange
rf.store	TEXT		StringRange
msc.no	INTEGER		IntegerRange (1 .. 10)
msc.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
msc.scan	CHOICE		EnumRange (NONE BOX RASTER SPIRAL)
msc.scanNum	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 AUTO)
msc.targetVel	FLOAT		DoubleRange (0.00000 .. 20.00000)
msc.maxError	FLOAT		DoubleRange (0.10 .. 20.00)
msc.name	TEXT		StringRange
msc.execute	TEXT		StringRange
msc.store	TEXT		StringRange
mps.no	INTEGER		IntegerRange (1 .. 10)
mps.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
mps.elSlope	FLOAT		DoubleRange (0.000 .. 200.000)
mps.xelSlope	FLOAT		DoubleRange (0.000 .. 200.000)

name	*type*	*flags*	*range*
mps.elOffset	FLOAT		DoubleRange (-1.000 .. 1.000)
mps.xelOffset	FLOAT		DoubleRange (-1.000 .. 1.000)
mps.polRef	FLOAT		DoubleRange (0.000 .. 359.999)
mps.elPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
mps.xelPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
mps.scanMode	CHOICE		EnumRange (OFF CONSTANT RANDOM)
mps.phaseMode	CHOICE		EnumRange (MANUAL TABLE)
mps.polErrorRot	CHOICE		EnumRange (OFF ON)
mps.execute	TEXT		StringRange
mps.store	TEXT		StringRange
pst.no	INTEGER		IntegerRange (1 .. 10)
pst.element	INTEGER		IntegerRange (1 .. 10)
pst.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
pst.bit0	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit1	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit2	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit3	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit4	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit5	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit6	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit7	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.elPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.azPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.polPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.store	TEXT		StringRange
profile	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
prf.select	CHOICE		EnumRange (0 - None)
prf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prf.no	INTEGER		IntegerRange (1 .. 20)
prf.name	TEXT		StringRange
prf.pointMode	CHOICE		EnumRange (STOP POS DESIGNATE PRESET POSITION GEO PRESET INTELSAT TRACK GEO DESIGNATE MANUAL POSITION STOW OPTRACK STAR TRACK TABLE TRACK NORAD TRACK MANUAL RATE SUN TRACK MOON TRACK POL AUTO)
prf.pointData	INTEGER		IntegerRange (0 .. 0)
prf.acquireMode	CHOICE		EnumRange (NONE BOX SCAN GEO SCAN)
prf.acquireData	INTEGER		IntegerRange (0 .. 11)
prf.trackMode	CHOICE		EnumRange (unknown NONE STEPTRACK MONOPULSE AUTOPHASE)
prf.trackData	INTEGER		IntegerRange (0 .. 11)
prf.rfData	INTEGER		IntegerRange (0 .. 20)
prf.configData	INTEGER		IntegerRange (0 .. 20)
prf.polMode	CHOICE		EnumRange (unknown STOP PRE MANRATE MAN POS OPTRACK INTLSAT TABTRAK MONOPULSE)
prf.polData	INTEGER		IntegerRange (0 .. 10)
prf.userInfo	TEXT		StringRange
prf.save	TEXT		StringRange
prf.activate	TEXT		StringRange
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ CW Final Limit)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Final Limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ CW Prelimit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Prelimit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Stow Pin Extended)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ Current Clamped)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Brake Fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ Brake CB Off)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Control 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Handcrank)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Overtemp)
faults.12	ALARM	R/O	AlarmFlagRange R/O (AZ CB Off)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ Motor 1 Rate)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL Up Limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (EL Down Limit)
faults.16	ALARM	R/O	AlarmFlagRange R/O (EL Up Prelimit)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL Down Prelimit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL Stow Pin Extended)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EL Current Clamped)
faults.20	ALARM	R/O	AlarmFlagRange R/O (EL Brake Fault)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (EL Brake CB Off)
faults.22	ALARM	R/O	AlarmFlagRange R/O (EL Motor Control 1)
faults.23	ALARM	R/O	AlarmFlagRange R/O (EL Motor Handcrank)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EL Motor Overtemp)
faults.25	ALARM	R/O	AlarmFlagRange R/O (EL CB Off)
faults.26	ALARM	R/O	AlarmFlagRange R/O (EL Motor 1 Rate)
faults.27	ALARM	R/O	AlarmFlagRange R/O (POL CW Limit)
faults.28	ALARM	R/O	AlarmFlagRange R/O (POL CCW Limit)
faults.29	ALARM	R/O	AlarmFlagRange R/O (POL CW Prelimit)
faults.30	ALARM	R/O	AlarmFlagRange R/O (POL CCW Prelimit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (POL Motor 1 Handcrank)
faults.32	ALARM	R/O	AlarmFlagRange R/O (POL Motor 2 Handcrank)
faults.33	ALARM	R/O	AlarmFlagRange R/O (POL Axis Stalled)
faults.34	ALARM	R/O	AlarmFlagRange R/O (POL Brake Fault)
faults.35	ALARM	R/O	AlarmFlagRange R/O (POL Brake CB Off)
faults.36	ALARM	R/O	AlarmFlagRange R/O (POL CB Off)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Interlock Logic Mismatch)
faults.38	ALARM	R/O	AlarmFlagRange R/O (ACU Stop Switch)
faults.39	ALARM	R/O	AlarmFlagRange R/O (ACU2 Stop Switch)

name	*type*	*flags*	*range*
faults.40	ALARM	R/O	AlarmFlagRange R/O (Antenna Hub Emergency)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Antenna Base Emergency)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Stairway Emergency)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Platform Emergency)
faults.44	ALARM	R/O	AlarmFlagRange R/O (3 Phase Fault)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Brake XFMR CB Off)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Heater CB Off)
faults.47	ALARM	R/O	AlarmFlagRange R/O (AZ Oscillation)
faults.48	ALARM	R/O	AlarmFlagRange R/O (EL Oscillation)
faults.49	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder)
faults.50	ALARM	R/O	AlarmFlagRange R/O (EL Encoder)
faults.51	ALARM	R/O	AlarmFlagRange R/O (AZ Tach/Encoder)
faults.52	ALARM	R/O	AlarmFlagRange R/O (EL Tach/Encoder)
faults.53	ALARM	R/O	AlarmFlagRange R/O (AZ Autotrack Runaway)
faults.54	ALARM	R/O	AlarmFlagRange R/O (EL Autotrack Runaway)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Low Signal)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Unacquirable signal)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Travel Limit AZ CW)

name	*type*	*flags*	*range*
faults.59	ALARM	R/O	AlarmFlagRange R/O (Travel Limit AZ CCW)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Travel Limit EL Up)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Travel Limit EL Down)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Travel Limit PL Up)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Travel Limit PL Down)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Monopulse Phasing)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Monopulse Bandwidth)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Tracking LNA)
faults.67	ALARM	R/O	AlarmFlagRange R/O (TRU BDC 1)
faults.68	ALARM	R/O	AlarmFlagRange R/O (TRU BDC 2)
faults.69	ALARM	R/O	AlarmFlagRange R/O (TRU BDC 3)
faults.70	ALARM	R/O	AlarmFlagRange R/O (TRU BDC 4)
faults.71	ALARM	R/O	AlarmFlagRange R/O (TRU DC Power)
faults.72	ALARM	R/O	AlarmFlagRange R/O (TRU Temperature)
faults.73	ALARM	R/O	AlarmFlagRange R/O (TRU Tracking)
faults.74	ALARM	R/O	AlarmFlagRange R/O (TRU RF Synth unlock)
faults.75	ALARM	R/O	AlarmFlagRange R/O (TRU Receiver unlock)
faults.76	ALARM	R/O	AlarmFlagRange R/O (TRU IF Synth unlock)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Receiver Datalink)

9.260 GDSatcom-ACU980

This page describes the device driver and the device window for the GDSatcom antenna controller ACU980. The driver supports the monopulse, OPTrack and steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the preset position memories of the ACU, lets you investigate the contents of the preset memories, store edited presets and execute presets.
- --- The third parameter page displays the tracking parameters and allows to start / stop resume OPTrack and steptrack tracking. The beacon receiver related parameters also are located at this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
hasPol2Axis	Defines if the antenna is equipped with a second polarization axis
hasTiltxis	Defines if the antenna is equipped with a tilt axis
hasTru	Defines it a GDSatcom TRU DTR is connected to the ACU
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The GDSatcom ACU 980 is controlled through a TCP connection to one of the ports 5001 or 5002 of the device. Each port only accepts one connection at a time. The communication has to be configured as to a serial terminal server port, specifying the device address and port

number in the device configuration.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.versionAcb	TEXT	R/O	StringRange R/O
info.versionFpga	TEXT	R/O	StringRange R/O
info.versionDsp	TEXT	R/O	StringRange R/O
info.diameter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.hasPol2Axis	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasTiltAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasTru	CHOICE	SAVE SETUP	EnumRange (NO ONE TWO)
internal.azLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plHigh	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plLow	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.pl2High	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.pl2Low	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.destUpdated	TEXT	R/O	StringRange R/O
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.19	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.20	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.21	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.22	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.23	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.24	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.25	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.26	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.27	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.28	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.29	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.30	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.31	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.32	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.33	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.34	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.35	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.36	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.37	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.38	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.39	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.40	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.41	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.42	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.43	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.44	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.45	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.46	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.47	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.48	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.49	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.50	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.51	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.52	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.53	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.54	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.55	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.56	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.57	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.58	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.59	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.60	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.61	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.62	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.63	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.64	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.65	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.66	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.67	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.68	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.69	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.70	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.71	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.72	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.73	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.74	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.75	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.76	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.77	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.78	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.79	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.80	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.81	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.84	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.85	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.86	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.89	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.90	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.91	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.92	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.93	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.94	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.95	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.96	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.97	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.98	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.99	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.100	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.101	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.102	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.103	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.104	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.105	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.106	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.107	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.108	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.109	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.110	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.111	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.112	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.113	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.114	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.115	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.116	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.117	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.118	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.119	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.120	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.121	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.122	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.123	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.124	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.125	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.126	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.127	HEX	R/O	HexRange R/O (0 .. 0)
internal.prf1	TEXT	R/O	StringRange R/O
internal.prf2	TEXT	R/O	StringRange R/O
internal.trstat.1	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.2	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.3	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.4	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.trstat.5	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.6	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.7	HEX	R/O	HexRange R/O (0 .. 0)
internal.trstat.8	HEX	R/O	HexRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
tilt	FLOAT		DoubleRange DIS (0.00 .. 0.00)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.tilt	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.mode3	TEXT	R/O	StringRange R/O
state.mode4	TEXT	R/O	StringRange R/O
state.submode1	TEXT	R/O	StringRange R/O
state.submode2	TEXT	R/O	StringRange R/O
state.submode3	TEXT	R/O	StringRange R/O
state.submode4	TEXT	R/O	StringRange R/O
state.lock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
state.xPos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.yPos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.elSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.polSlope	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.polPhase	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.elError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.polError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.scanMode	TEXT	R/O	StringRange R/O
state.phaseMode	TEXT	R/O	StringRange R/O
state.rfSwitch	TEXT	R/O	StringRange R/O
state.autoAlg	TEXT	R/O	StringRange R/O
state.sumRhcp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sumLhcp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sumChannel	CHOICE	R/O	EnumRange R/O (UNKNOWN RHCP LHCP V H)
state.errChannel	CHOICE	R/O	EnumRange R/O (UNKNOWN RHCP LHCP)
state.lp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.cp	BOOLEAN	R/O	BooleanRange R/O (true false)
state.cpLp	CHOICE	R/O	EnumRange R/O (UNKNOWN LINEAR CIRCULAR)
prespos.no	INTEGER		IntegerRange (1 .. 40)

name	*type*	*flags*	*range*
prespos.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prespos.az	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.el	FLOAT		DoubleRange (0.000 .. 0.000)
prespos.pl	FLOAT		DoubleRange (0.00 .. 0.00)
prespos.pl2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
prespos.name	TEXT		StringRange
prespos.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
prespos.cw	CHOICE		EnumRange (CLOCKWISE COUNTERCLOCKWISE SHORTEST)
prespos.execute	TEXT		StringRange
prespos.store	TEXT		StringRange
optrack.no	INTEGER		IntegerRange (1 .. 3)
optrack.name	TEXT		StringRange
optrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
optrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
optrack.sct	CHOICE		EnumRange (10 min 20 min 30 min)
optrack.config	CHOICE		EnumRange (NONE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
optrack.type	CHOICE		EnumRange (STEPTRACK MONOPULSE)
optrack.axis	CHOICE		EnumRange (ALL AZ/EL)
optrack.execute	TEXT		StringRange
optrack.store	TEXT		StringRange
steptrack.no	INTEGER		IntegerRange (1 .. 10)
steptrack.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
steptrack.recycle	FLOAT		DoubleRange (-50.0 .. 0.0)
steptrack.snt	INTEGER		IntegerRange (1 .. 4)

name	*type*	*flags*	*range*
steptrack.size	FLOAT		DoubleRange (5.0 .. 20.0)
steptrack.data	CHOICE		EnumRange (OFF ON)
steptrack.sct	FLOAT		DoubleRange (0.0 .. 240.0)
steptrack.gain	FLOAT		DoubleRange (0.00 .. 1.00)
steptrack.scan	CHOICE		EnumRange (NO SCAN BOX SCAN RASTER SCAN SPIRAL SCAN GEO SCAN)
steptrack.scanNum	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 AUTO)
steptrack.park	FLOAT		DoubleRange (3.0 .. 10.0)
steptrack.limit	FLOAT		DoubleRange (0.0 .. 10.0)
steptrack.incl	FLOAT		DoubleRange (0.0 .. 15.0)
steptrack.name	TEXT		StringRange
steptrack.execute	TEXT		StringRange
steptrack.store	TEXT		StringRange
steptrack.cancel	TEXT		StringRange
steptrack.tracknow	TEXT		StringRange
cfg.no	INTEGER		IntegerRange (1 .. 20)
cfg.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
cfg.acqThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.sigThreshold	FLOAT		DoubleRange (-99.9 .. 0.0)
cfg.name	TEXT		StringRange
cfg.execute	TEXT		StringRange
cfg.store	TEXT		StringRange
rf.no	INTEGER		IntegerRange (1 .. 20)
rf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
rf.frequency	FLOAT		DoubleRange (100.000 .. 100000.000)
rf.bandwidth	CHOICE		EnumRange (6 kHz 280 kHz)

name	*type*	*flags*	*range*
rf.slope	FLOAT		DoubleRange (0.1000 .. 100.0000)
rf.offset	FLOAT		DoubleRange (-100.0000 .. 100.0000)
rf.activeSearch	CHOICE		EnumRange (OFF ON)
rf.feedOffsetEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.feedOffsetXEl	FLOAT		DoubleRange (-5.000 .. 5.000)
rf.path	CHOICE		EnumRange (1 2)
rf.sweepWidth	FLOAT		DoubleRange (40.0 .. 145.0)
rf.secondaryPath	INTEGER		IntegerRange DIS (0 .. 2)
rf.name	TEXT		StringRange
rf.bdcFrequency	FLOAT		DoubleRange (0.000 .. 100000.000)
rf.execute	TEXT		StringRange
rf.store	TEXT		StringRange
rf.polSelect	CHOICE		EnumRange (LHCP RHCP H V)
rf.polMode	CHOICE		EnumRange (CIRCULAR LINEAR)
rf.pol2Mode	CHOICE		EnumRange DIS (CIRCULAR LINEAR)
msc.no	INTEGER		IntegerRange (1 .. 10)
msc.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
msc.scan	CHOICE		EnumRange (NONE BOX RASTER SPIRAL)
msc.scanNum	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 AUTO)
msc.targetVel	FLOAT		DoubleRange (0.00000 .. 20.00000)
msc.maxError	FLOAT		DoubleRange (0.10 .. 20.00)
msc.name	TEXT		StringRange
msc.execute	TEXT		StringRange
msc.store	TEXT		StringRange
mps.no	INTEGER		IntegerRange (1 .. 10)

name	*type*	*flags*	*range*
mps.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
mps.elSlope	FLOAT		DoubleRange (0.000 .. 200.000)
mps.xelSlope	FLOAT		DoubleRange (0.000 .. 200.000)
mps.elOffset	FLOAT		DoubleRange (-1.000 .. 1.000)
mps.xelOffset	FLOAT		DoubleRange (-1.000 .. 1.000)
mps.polRef	FLOAT		DoubleRange (0.000 .. 359.999)
mps.elPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
mps.xelPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
mps.scanMode	CHOICE		EnumRange (OFF CONSTANT RANDOM)
mps.phaseMode	CHOICE		EnumRange (MANUAL TABLE)
mps.polErrorRot	CHOICE		EnumRange (OFF ON)
mps.execute	TEXT		StringRange
mps.store	TEXT		StringRange
pst.no	INTEGER		IntegerRange (1 .. 10)
pst.element	INTEGER		IntegerRange (1 .. 10)
pst.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
pst.bit0	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit1	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit2	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit3	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit4	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit5	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit6	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit7	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit8	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit9	CHOICE		EnumRange (OPEN CLOSED IGNORE)

name	*type*	*flags*	*range*
pst.bit10	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit11	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit12	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit13	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit14	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.bit15	CHOICE		EnumRange (OPEN CLOSED IGNORE)
pst.elPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.azPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.polPhaseShift	FLOAT		DoubleRange (0.000 .. 359.999)
pst.store	TEXT		StringRange
profile	TEXT	R/O	StringRange R/O
prf.select	CHOICE		EnumRange (0 - None)
prf.edited	BOOLEAN	R/O	BooleanRange R/O (true false)
prf.no	INTEGER		IntegerRange (1 .. 20)
prf.name	TEXT		StringRange
prf.pointMode	CHOICE		EnumRange (STOP POS DESIGNATE PRESET POSITION GEO PRESET INTELSAT TRACK GEO DESIGNATE MANUAL POSITION STOW OPTRACK STAR TRACK TABLE TRACK NORAD TRACK MANUAL RATE SUN TRACK MOON TRACK POL AUTO)
prf.pointData	INTEGER		IntegerRange (0 .. 0)
prf.acquireMode	CHOICE		EnumRange (NONE BOX SCAN GEO SCAN)
prf.acquireData	INTEGER		IntegerRange (0 .. 11)
prf.trackMode	CHOICE		EnumRange (unknown NONE STEPTRACK OPTRACK MONOPULSE AUTOPHASE)
prf.trackData	INTEGER		IntegerRange (0 .. 11)

name	*type*	*flags*	*range*
prf.rfData	INTEGER		IntegerRange (0 .. 20)
prf.rfData2	INTEGER		IntegerRange (0 .. 20)
prf.configData	INTEGER		IntegerRange (0 .. 20)
prf.polMode	CHOICE		EnumRange (unknown STOP PRE MANRATE MAN POS OPTRACK INTLSAT TABTRAK AUTO)
prf.polData	INTEGER		IntegerRange (0 .. 10)
prf.pol2Mode	CHOICE		EnumRange (unknown STOP PRE MANRATE MAN POS OPTRACK INTLSAT TABTRAK AUTO)
prf.pol2Data	INTEGER		IntegerRange (0 .. 10)
prf.userInfo	TEXT		StringRange
prf.save	TEXT		StringRange
prf.activate	TEXT		StringRange
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ CW Limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ CW Prelimit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Prelimit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Stow PIN)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ Brake)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Brake CB Off)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ Axis Summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Control)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Handcrank)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Overtemp)
faults.12	ALARM	R/O	AlarmFlagRange R/O (AZ Motor 3-Phase OFF)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Rate)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL Up Limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (EL Down Limit)
faults.16	ALARM	R/O	AlarmFlagRange R/O (EL Up Prelimit)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL Down Prelimit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (El Stow Pin)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EL Brake)
faults.20	ALARM	R/O	AlarmFlagRange R/O (El Brake CB Off)
faults.21	ALARM	R/O	AlarmFlagRange R/O (El Axis Summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (El Motor Control)
faults.23	ALARM	R/O	AlarmFlagRange R/O (EL Motor Handcrank)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EL Motor Overtemp)
faults.25	ALARM	R/O	AlarmFlagRange R/O (El Motor 3-Phase OFF)
faults.26	ALARM	R/O	AlarmFlagRange R/O (El Motor Rate)
faults.27	ALARM	R/O	AlarmFlagRange R/O (POL CW Limit)
faults.28	ALARM	R/O	AlarmFlagRange R/O (POL CCW Limit)
faults.29	ALARM	R/O	AlarmFlagRange R/O (POL CW Prelimit)
faults.30	ALARM	R/O	AlarmFlagRange R/O (POL CCW Preimit)

name	*type*	*flags*	*range*
faults.31	ALARM	R/O	AlarmFlagRange R/O (POL Motor Handcrank)
faults.32	ALARM	R/O	AlarmFlagRange R/O (POL Stall)
faults.33	ALARM	R/O	AlarmFlagRange R/O (POL Stabilizer)
faults.34	ALARM	R/O	AlarmFlagRange R/O (POL Brake)
faults.35	ALARM	R/O	AlarmFlagRange R/O (POL Brake CB)
faults.36	ALARM	R/O	AlarmFlagRange R/O (POL Motor Control)
faults.37	ALARM	R/O	AlarmFlagRange R/O (POL Motor Overtemp)
faults.38	ALARM	R/O	AlarmFlagRange R/O (POL Motor 3-Phase Off)
faults.39	ALARM	R/O	AlarmFlagRange R/O (POL Motor Rate)
faults.40	ALARM	R/O	AlarmFlagRange R/O (POL Axis)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Interlock Logic)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Safety Circuit)
faults.43	ALARM	R/O	AlarmFlagRange R/O (AZ Resolver)
faults.44	ALARM	R/O	AlarmFlagRange R/O (EL Resolver)
faults.45	ALARM	R/O	AlarmFlagRange R/O (ACU Stop Switch)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Expansion Board)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop ACU)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop Pedestal)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop Hub)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop Ladder)
faults.51	ALARM	R/O	AlarmFlagRange R/O (System Fault)

name	*type*	*flags*	*range*
faults.52	ALARM	R/O	AlarmFlagRange R/O (3-Phase fault)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Brake XFMR CB off)
faults.54	ALARM	R/O	AlarmFlagRange R/O (EL Low Angle Cutoff)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Air Cond. CB Off)
faults.56	ALARM	R/O	AlarmFlagRange R/O (PDU Heater CB Off)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.58	ALARM	R/O	AlarmFlagRange R/O (CPU Watchdog Timeout)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Red. FPGA DSER Hot)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Main FPGA DSER Hot)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Expansion Board)
faults.62	ALARM	R/O	AlarmFlagRange R/O (EL Aux Drive)
faults.63	ALARM	R/O	AlarmFlagRange R/O (AZ Aux Drive)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Simulation CCU ON)
faults.65	ALARM	R/O	AlarmFlagRange R/O (AZ Oscillation)
faults.66	ALARM	R/O	AlarmFlagRange R/O (EL Oscillation)
faults.67	ALARM	R/O	AlarmFlagRange R/O (POL Oscillation)
faults.68	ALARM	R/O	AlarmFlagRange R/O (TILT Oscillation)
faults.69	ALARM	R/O	AlarmFlagRange R/O (AZ Encoder)
faults.70	ALARM	R/O	AlarmFlagRange R/O (EL Encoder)
faults.71	ALARM	R/O	AlarmFlagRange R/O (AZ Tach)
faults.72	ALARM	R/O	AlarmFlagRange R/O (EL Tach)

name	*type*	*flags*	*range*
faults.73	ALARM	R/O	AlarmFlagRange R/O (DSP)
faults.74	ALARM	R/O	AlarmFlagRange R/O (AZ Velocity)
faults.75	ALARM	R/O	AlarmFlagRange R/O (EL Velocity)
faults.76	ALARM	R/O	AlarmFlagRange R/O (POL Velocity)
faults.77	ALARM	R/O	AlarmFlagRange R/O (TILT Velocity)
faults.78	ALARM	R/O	AlarmFlagRange R/O (AZ Acceleration)
faults.79	ALARM	R/O	AlarmFlagRange R/O (EL Acceleration)
faults.80	ALARM	R/O	AlarmFlagRange R/O (POL Acceleration)
faults.81	ALARM	R/O	AlarmFlagRange R/O (TILT Acceleration)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Region AZ CW)
faults.83	ALARM	R/O	AlarmFlagRange R/O (Region AZ CCW)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Region EL UP)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Region EL DOWN)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Region PL UP)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Region PL DOWN)
faults.88	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CW)
faults.89	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CCW)
faults.90	ALARM	R/O	AlarmFlagRange R/O (SW Limit EL Up)
faults.91	ALARM	R/O	AlarmFlagRange R/O (SW Limit EL Down)
faults.92	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL CW)

name	*type*	*flags*	*range*
faults.93	ALARM	R/O	AlarmFlagRange R/O (SW Limit PL CCW)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Box AZ/EL)
faults.95	ALARM	R/O	AlarmFlagRange R/O (Latitude Limit)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Longitude Limit)
faults.97	ALARM	R/O	AlarmFlagRange R/O (Sun Outage)
faults.98	ALARM	R/O	AlarmFlagRange R/O (Monopulse Phasing)
faults.100	ALARM	R/O	AlarmFlagRange R/O (Simulation ACU On)
faults.101	ALARM	R/O	AlarmFlagRange R/O (Axis Options)
faults.102	ALARM	R/O	AlarmFlagRange R/O (POL/TILT LP/CP)
faults.103	ALARM	R/O	AlarmFlagRange R/O (Board Number)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Loop Test)
faults.105	ALARM	R/O	AlarmFlagRange R/O (ACB Link Down)
faults.106	ALARM	R/O	AlarmFlagRange R/O (Receiver Datalink)
faults.107	ALARM	R/O	AlarmFlagRange R/O (Config File Change)
faults.108	ALARM	R/O	AlarmFlagRange R/O (Link Down Summary)
faults.109	ALARM	R/O	AlarmFlagRange R/O (Low Signal)
faults.110	ALARM	R/O	AlarmFlagRange R/O (Unacquirable Signal)
faults.111	ALARM	R/O	AlarmFlagRange R/O (Bad Track Point)
faults.112	ALARM	R/O	AlarmFlagRange R/O (High Signal)
faults.113	ALARM	R/O	AlarmFlagRange R/O (Monopulse Bandwidth)
faults.114	ALARM	R/O	AlarmFlagRange R/O (Load Site Location)

name	*type*	*flags*	*range*
faults.115	ALARM	R/O	AlarmFlagRange R/O (Low-Level)
faults.116	ALARM	R/O	AlarmFlagRange R/O (Others Task)
faults.117	ALARM	R/O	AlarmFlagRange R/O (Disable AZ Key)
faults.118	ALARM	R/O	AlarmFlagRange R/O (Disable EL Key)
faults.119	ALARM	R/O	AlarmFlagRange R/O (Disable PL Key)
faults.120	ALARM	R/O	AlarmFlagRange R/O (Optrack 1/2/3)
faults.123	ALARM	R/O	AlarmFlagRange R/O (PX)
faults.124	ALARM	R/O	AlarmFlagRange R/O (LAN/WAN MAC Address)
faults.125	ALARM	R/O	AlarmFlagRange R/O (Real Time Clock)
faults.126	ALARM	R/O	AlarmFlagRange R/O (TRU Switch)
faults.127	ALARM	R/O	AlarmFlagRange R/O (ACB Temp)
faults.128	ALARM	R/O	AlarmFlagRange R/O (RCVR)
faults.129	ALARM	R/O	AlarmFlagRange R/O (RCVR Thresh. err. AZ/El)
faults.130	ALARM	R/O	AlarmFlagRange R/O (PLC)
faults.131	ALARM	R/O	AlarmFlagRange R/O (Obstruction)
faults.132	ALARM	R/O	AlarmFlagRange R/O (Obscura Limit)
faults.133	ALARM	R/O	AlarmFlagRange R/O (Pol Stop Mode)
faults.134	ALARM	R/O	AlarmFlagRange R/O (ACB Voltage)
faults.135	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal AZ)
faults.136	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal EL)
faults.137	ALARM	R/O	AlarmFlagRange R/O (Resolver Cal PL)
faults.138	ALARM	R/O	AlarmFlagRange R/O (Travel/SW Limit TILT)
faults.140	ALARM	R/O	AlarmFlagRange R/O (FPGA Config)
faults.141	ALARM	R/O	AlarmFlagRange R/O (Disable Tilt Key)
faults.142	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.261 GDSatcom-DTR520

This page describes the device driver and the device window for the for the GDSatcom Digital Tracking Receiver Model 520/550 using the GDsatcom-TRU protocol.

The default TCP/IP communication port is : 5060

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The beacon receiver parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The Management IP address specification for the device followed by the TCP/IP Port. e.g.: 10.10.1.10:5060
aliasPolA	Defines the polarization A.
aliasPolB	Defines the polarization B.
controlMode	Defines the control mode local or remote.
vcoMode	Defines the VCO mode automatic or manual.
vcoOffset	Defines the VCO offset in kHz.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.fpgaVersion	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.trackMode	CHOICE		EnumRange (AZ/EL/POL MONOPULSE AZ/EL MONOPULSE POL MONOPULSE STEPTRACK)
config.aliasPolA	TEXT	SAVE SETUP	StringRange
config.aliasPolB	TEXT	SAVE SETUP	StringRange
config.controlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.vcoMode	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.vcoOffset	FLOAT	SETUP	DoubleRange (-99.9 .. 99.9)
config.model	CHOICE	SAVE SETUP	EnumRange (DTR520 DTR550)
power	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
relPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rfMode	CHOICE	R/O	EnumRange R/O (DIGITAL ANALOG WIDEBAND DVB)
frequency	FLOAT		DoubleRange (890.000 .. 30000.000)
attenuation	INTEGER		IntegerRange (0 .. 31)

name	*type*	*flags*	*range*
sweepWidth	INTEGER		IntegerRange (8 .. 500)
polarization	CHOICE		EnumRange (A B)
bandwidth	CHOICE		EnumRange (0 - NONE)
beaconControl	CHOICE		EnumRange (CW 800HZBPSK)
signalOffset	FLOAT		DoubleRange (-100.0 .. 200.0)
reset	TEXT	NOPRESET	StringRange
internal.range	TEXT	R/O	StringRange R/O
internal.pol	CHOICE	R/O	EnumRange R/O (0 1)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.availBandwidth	TEXT	R/O	StringRange R/O
internal.bw1	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw2	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw3	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw4	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw5	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw6	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw7	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw8	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.bw1text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw2text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw3text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw4text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.bw5text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw6text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw7text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.bw8text	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.vcoMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary Fault Active)
faults.03	ALARM	R/O	AlarmFlagRange R/O (VCO Auto Control)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Track Fault Active)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature Fault)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF Synth. Unlocked)
faults.08	ALARM	R/O	AlarmFlagRange R/O (IF Synth. Unlocked)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PLL Unlocked)
faults.10	ALARM	R/O	AlarmFlagRange R/O (V1 Power Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (V2 Power Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (V3 Power Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RCVR Synth. Unlocked)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Track Mode)

9.262 GDSatcom-RSC1200

This page describes the device driver and the device window for the GDSatcom LNA/LNB Redundancy Controller RSC-1200.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the redundancy settings
- --- The second page shows the measurement values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
currentWindow	Current deviation which is acceptable without issue an alarm
latchedFaults	Enable/Disable latched faults

Remarks

none

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitType	CHOICE	R/O	EnumRange R/O (1:1 DUAL-1:1 1:2)
info.version	TEXT	R/O	StringRange R/O
info.hasTrackingUnit	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.nominalCurrent.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
info.nominalCurrent.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
info.nominalCurrent.3	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
info.nominalCurrent.4	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
info.nominalCurrent.5	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.currentWindow	CHOICE	SETUP	EnumRange (5% 10% 15% 20% 25%)
config.latchedFaults	CHOICE	SETUP	EnumRange (ON OFF)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.5	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
protection	CHOICE		EnumRange (MANUAL AUTO)
pol1	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
pol2	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
position	CHOICE		EnumRange (2-STBY 1-STBY 3-STBY A B)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
priority	CHOICE		EnumRange (NONE POL1 POL2)
priority.2	CHOICE		EnumRange (NONE UNIT1 UNIT2)
revert	CHOICE		EnumRange (ENABLED DISABLED)
resetFaults	TEXT		StringRange

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PS Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 (Track))
faults.16	ALARM	R/O	AlarmFlagRange R/O (Latched Unit Summary)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Latched Switch)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Latched PS Summary)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Latched Controller)
internal.v1	TEXT	R/O	StringRange R/O
internal.v2	TEXT	R/O	StringRange R/O
internal.v3	TEXT	R/O	StringRange R/O

9.263 GDSatcom-Rack-SSPA

This page describes the device driver and the device window for the for the GDSatcom Single and Dual Module Rack SSPAs via Serial I/O using the GDSatcom-STX protocol.

It was tested with a SMR (Single Module)

- Board Version: 1.93

- Module Version: 3.07

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters.
- --- the module and transmit meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The Serial Device Address. Factory Default is 1.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
model	Sets the model SMR or DMR which should be controlled with the device.
module.1.enable	Defines the 1st module enable state.
module.2.enable	Defines the 2nd module enable state.
powerUpState	Defines the units transmit state after a power cycle.
units	Defines the measurement units displayed locally on the Front Panel which then also apply to the remote interface dBm,dBW or W.
refPwrMeas	Defines the desired reflected power measurement method.
localLockout	Defines the local lockout state.
limitLow	Defines the low rf level limit.
limitHigh	Defines the high rf level limit.
limit.Refl	Defines the reflected level limit.

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*

9.264 GDSatcom-SSPA-QuadMod

This page describes the device driver and the device window for the for the GDSatcom QuadMod SSPAs via Serial I/O using the GDSatcom-STX protocol.

It was tested with:

- Board Version 2.31
- Module Version 3.07

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters.
- --- the module and transmit meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The Serial Device Address. Factory Default is 1.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

parameter	*description*
powerUpState	Defines the units transmit state after a power cycle.
units	Defines the measurement units displayed locally on the Front Panel which then also apply to the remote interface dBm,dBW or W.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.amptype	TEXT	R/O	StringRange R/O
info.version.board	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
info.version.module1	TEXT	R/O	StringRange R/O
info.version.module2	TEXT	R/O	StringRange R/O
info.version.module3	TEXT	R/O	StringRange R/O
info.range	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.limitLow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.limitHigh	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.limitRefl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.hasParallel	TEXT	R/O	StringRange R/O
info.hasRemoteParallel	TEXT	R/O	StringRange R/O
info.hasSwitch	TEXT	R/O	StringRange R/O
info.hasDualPS	TEXT	R/O	StringRange R/O
info.measRef	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.powerUpState	CHOICE	SETUP	EnumRange (MUTE OPERATE LAST)
config.units	CHOICE	SETUP	EnumRange (dBm dBW W)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (-20.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
meas.current4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageRegulated1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageGate1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageInput1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageRegulated2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageGate2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageInput2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageRegulated3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageInput3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageGate3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageRegulated4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageInput4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltageGate4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.power1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.power2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.power3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.power4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.module1	TEXT	R/O	StringRange R/O
state.module2	TEXT	R/O	StringRange R/O
state.module3	TEXT	R/O	StringRange R/O
state.module4	TEXT	R/O	StringRange R/O
operatingTime	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
position	CHOICE		EnumRange DIS (A B)
location	CHOICE		EnumRange DIS (ANTENNA LOAD)
internal.days	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.seconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gainRange	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit Fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit Warning)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PS Disconnect)
faults.04	ALARM	R/O	AlarmFlagRange R/O (High Fwd Power)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low Fwd Power)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reflected Power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Logic Board)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PSU Module 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU Module 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU Module 3)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 1A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 2A)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 3A)

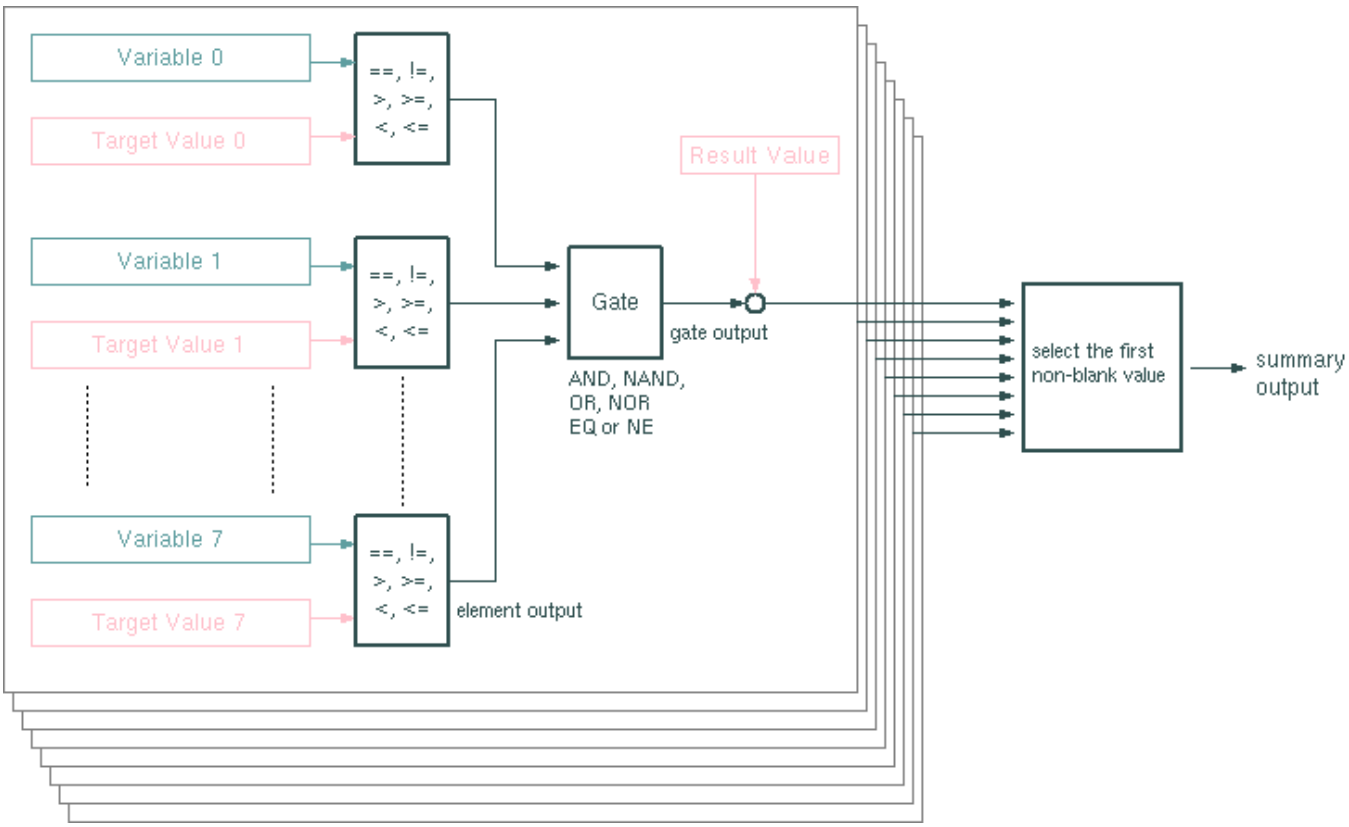
name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 1B)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 2B)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Dual PSU Module 3B)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Module PSU mismatch)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Module 1)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Module 2)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Module 3)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Module 4)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 1)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 2)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 3)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 4)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 5)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 6)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 7)

name	*type*	*flags*	*range*
faults.34	ALARM	R/O	AlarmFlagRange R/O (Parallel Input 8)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Front Panel)
faults.36	ALARM	R/O	AlarmFlagRange R/O (PIO)
faults.37	ALARM	R/O	AlarmFlagRange R/O (ID ROM Memory)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Network)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Fwd Pwr Detector)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Refl Pwr Detector)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Module Disabled)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Local Mode)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Maintenance Mode)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Power Up)
faults.45	ALARM	R/O	AlarmFlagRange R/O (New Fault)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 1)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 2)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 3)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 4)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 5)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 6)

name	*type*	*flags*	*range*
faults.52	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 7)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Remote Parallel Input 8)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Local Mute)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Serial IO Mute)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Parallel IO Mute)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Network Mute)
faults.58	ALARM	R/O	AlarmFlagRange R/O (RF Inhibit)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Switch Inhibit)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Modules Off)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Modules Disabled)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Switch)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.265 Gate-Array

The Gate-Array and Gate-Array-20 devices provide a set of 8 logical gates with 8 inputs each (20 with the Gate-Array-20) which may be used to merge status variables in a user programmed way. Gate-Array and Gate-Array-20 behave exactly the same way, the only difference is the number of inputs of each gate. Each gate may be configured to work as a AND, NAND, OR, NOR, NE (XOR) or EQ gate. The inputs of a gate are so called elements, which compare the contents of a sat-nms parameter to a fixed target value.



Elements

The Gate-Array altogether provides 64 'elements' (160 with the Gate-Array-20) which compare a parameter value to a fixed target value. Each element of a Gate-Array may be used / programmed individually.

The output of an element is a read-only text-type variable which either reads 'true' or 'false'. This output may may be monitored by other devices, the parameter name is ' *<device-name>.<gate-no.>.<element-no.>.output* '. Counting the gate or element numbers starts at zero, the name of the second element of the first gate e.g. is ' *<device-name>.0.1.output* '.

The parameter 'message id' defines the full name of the parameter the element shall monitor. The 'target' value is the constant value the parameter is compared against. The comparison operator is selected from a choice box:

- EQ --- 'equal' The output of the element becomes 'true', if the monitored parameter is *textual* equal to the target value. If you compare numeric values with EQ, you have to consider the textual representation of a parameter: '1.0' is not equal to '1.00'! The monitored parameter is read like it would appear in the user interface, but without a unit string.
- NE --- 'not equal' The inverted form of EQ.
- GT --- 'greater than' The output of the element becomes 'true', if the monitored parameter is *numerically* greater than the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.
- GE --- 'greater or equal' The output of the element becomes 'true', if the monitored

parameter is *numerically* greater or equal the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.
- LT --- 'less than' The output of the element becomes 'true', if the monitored parameter is *numerically* less than the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.
- LE --- 'less or equal' The output of the element becomes 'true', if the monitored parameter is *numerically* less or equal the target value. If any of the compared values cannot be successfully parsed as a floating point number, the value is interpreted as zero.

Elements which have no or no valid message ID set, are not taken into account in the downstream gate. The same applies to elements which indeed have a valid message ID set, but the message source did not supply a value since system start.

Gates

Every 8/20 elements are connected to one gate (There are 8 gates in total). Each gated can be set to perform one of the logical functions AND, NAND, OR, NOR, EQ and NE. A gate likewise provides an output variable, named ' *<device-name>.<gate-no.>.output* ' (' *<device-name>.0.output* ' designates the output of the first gate). A output either reads 'true' or 'false' depending on the result of the logical function it performs. For the logical function only these inputs contribute which really are in use. The outputs of elements which have no message identifier set are not taken into account. The available logical functions for the gate are:

- AND --- Output is *true* if all active input comparisons are true.
- NAND --- Output is *false* if all active input comparisons are true.
- OR --- Output is *true* if at least one of the active input comparisons is true.
- NOR --- Output is *false* if at least one of the active input comparisons is true.
- EQ --- Output is *true* if all active input comparisons result to the same logical state
- NE --- Output is *false* if all active input comparisons result to the same logical state

Beside this, each gate provides a parameter called 'result'. This parameter is an arbitrary text value which may become the summary output value of the gate array under certain circumstances.

The Summary Output

The Gate-Array provides a summary output variable named ' *<device-name>.output* '. This summary output carries the value of the 'result' parameter of the first gate which has a 'true' output. If no gate has a 'true' output, the Gate-Array's master output reads an empty string. Example: If the gates 4 and 6 in the Gate-Array are 'true', the summary output is set to the value of the 'result' variable of gate 4.

Individual Logical Gate Outputs

The Gate-Array provides variables which report the local state of each comparison element, each gate and additionally a summary output which combines the outputs of all gates with a logical OR. For compatibility reasons these variables all are read-only TEXT variables reporting either 'true' or 'false'.

On element level (the result of each individual comparison) the variables are named '*<device-name>.<gate-no.>.<element-no.>.output* '. Counting the gate or element numbers starts at zero, the name of the second element of the first gate e.g. is ' *<device-name>.0.1.output* '.

On gate level (the result of each of the eight gates) the variables are named '*<device-name>.<gate-no.>.output* '. Counting the gate numbers starts at zero.

Beside the textual summary output described above, the Gate-Array provides a second summary output called ' *<device-name>.ored* '. This output reports 'true' if at least one of the gates reports 'true', 'false' otherwise.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The configuration and status pages each contain the parameters for one gate and the eight elements which are connected to this gate. The actual state of the element outputs and the gate output is signalled by the color of the arrows which show the logical signal flow. Gray arrows signal a 'false' or inactive state, green arrows signal 'true' state.At the bottom of each page the actual summary (master) output of the Gate-Array and the output states of all eight gates ist shown.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

Beside the standard debug settings 'verbose' and 'logParameterChanges' there is one configuration parameter available for the Gate-Array logical device.

parameter	***description***
outputDefault	If this parameter is set to a non-blanc value, the Gate-Arrays 'output' shows this value instead of an empty string if none of the gates results 'true'.

Remarks

- The Gate-Array is particularly suitable to generate color selections for the user interface from complex status information in the M&C. For example, switch settings, rf-on states and fault signals may be combined to the information if a certain signal path is carrying signal, switched off or being faulty. The summary output of the Gate-Array may be monitored by the line elements in the user interface which symbolize this signal path. They will change their color following the detected state.
- The gates of a Gate-Array may be cascaded by referencing the output of one gate as the message ID of another's gate input element. Also 'feedback' routings are possible. If you couple element inputs to an output of the same gate array, you should consider, that a

gate's output changes with up to one second delay after the gate's input changed. If you cascade Gates, the delay times are accumulating. If you feed-back signals, oscillation due to the delay may be the result.
- The variables used to monitor and control a Gate-Array are created dynamically at runtime. The neither appear in the device driver file nor do the in the variable list below. For the same reason it is not possible to map the output variables of a Gate-Array into another M&C via a Proxy-Device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.outputDefault	TEXT	SAVE SETUP	StringRange

9.266 Generex-CS121-UPS-Adapter

This page describes the device driver and the device window for the Generex CS121 UPS SNMP/Web Adapter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the actual state of the UPS, presence of input power, battery capacity etc.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page the only parameter to be set is the IP address of the CS121 adapter. Please also notice the section "Prepare the CS121 for the use with the sat-nms software" below.

Fault monitoring

The device driver monitors a couple of fault conditions at the device. These are:

- faults.01 --- UPS Summary The summary fault state as reported from the device. This includes the more detailed fault conditions below, but also other device specific faults which depend on the type of the UPS monitored by the CS121. If this fault is solely active, the UPS should be inspected through the CS121 web browser interface in order to determine the kind of malfunction.
- faults.02 --- Mains input This fault indicates that the mains input is off / missing, the UPS is running on battery.
- faults.03 --- Low battery The battery capacity is low, the UPS may be unable to feed the output within the next few minutes.
- faults.04 --- Depleted battery The battery is empty, in case of an input power outage, the output will be off.

Please note, that the "Low battery" / "Depleted battery" faults appear exclusively, i.e. "Low battery" disappears if the battery becomes completely empty. The faults also may appear while the input is OK, indicating the possible consequence of an input power loss.

Prepare the CS121 for the use with the sat-nms software

The sat-nms software controls the CS121 using SNMP. For this, SNMP control must be enabled at the CS121. Please refer to the CS121 user manual for details, a summary of this procedure is listed below.

1. Set the network parameters (IP address / subnet mask / gateway) at the CS121 to make it accessible from the VLC.
2. With the CS121 web user interface, at the *Configuration / Network&Security* page, set the *Enable SNMP Agent* check mark to switch on SNMP in general. Press *Apply* to accept the change.
3. At the *Configuration / SNMP* page in the table *SNMP Communities* , add the IP address of the VLC controlling the CS121. Set the community for this address to *public* and the permission to *Read/Write* . Again press *Apply* to accept the change.
4. At the *Configuration / Save Configuration* page select *Save, Exit & Reboot* to make the changes effective. Be sure, that the UPS has mains power available during the reboot, it might hang otherwise.

Without having the VLC's IP address in the *SNMP Communities* table, the CS121 will silently ignore all communication requests from the VLC, the sat-nms user interface will show a communication fault (Read timeout) for the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
battery.state	TEXT	R/O	StringRange R/O
battery.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
battery.capacity	INTEGER	R/O	IntegerRange R/O (0 .. 100)
battery.since	INTEGER	R/O	IntegerRange R/O (0 .. 0)
battery.remaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.voltage	FLOAT	R/O	DoubleRange R/O (0 .. 0)
input.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.power	FLOAT	R/O	DoubleRange R/O (0 .. 0)
input.lineBads	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.source	TEXT	R/O	StringRange R/O
output.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.voltage	FLOAT	R/O	DoubleRange R/O (0 .. 0)
output.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.power	FLOAT	R/O	DoubleRange R/O (0 .. 0)
output.load	INTEGER	R/O	IntegerRange R/O (0 .. 100)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (UPS Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Mains input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low battery)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Depleted battery)

9.267 Generic

The Generic device is a dummy device driver implementation which only supports low level commands to be sent to the device. It may be used with device for which no device driver is available up to now.

The Generic device driver prefers the Terminal-CR-CRLF-E protocol (Messages sent are terminated with CR, the protocol expects a an echo sent by the device prior to the reply. All reply are terminated by a CRLF pair). You may, however, use any text based protocol with this device driver.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

9.268 Geosync-PS

This page describes the device driver and the device window for the Geosync Multi-Voltage Amplifier Power Supply System, through TCP Port 10001.

The driver was tested with the

- PAR4-3V-121520

and the following SW Releases:

- SW: REV1.00 04/22/14

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Displays the red power supplies voltages and amplifiers current measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	The address of the unit (Default = @)
iLowLimit.1	Defines the low current limit for Amplifier No. 1 in mA
iHighLimit.1	Defines the high current limit for Amplifier No. 1 in mA
iDeltaLimit.1	Defines the delta current limit for Amplifier No. 1 in mA
iLowLimit.2	Defines the low current limit for Amplifier No. 2 in mA
iHighLimit.2	Defines the high current limit for Amplifier No. 2 in mA
iDeltaLimit.2	Defines the delta current limit for Amplifier No. 2 in mA
iLowLimit.3	Defines the low current limit for Amplifier No. 3 in mA
iHighLimit.3	Defines the high current limit for Amplifier No. 3 in mA
iDeltaLimit.3	Defines the delta current limit for Amplifier No. 3 in mA
iLowLimit.4	Defines the low current limit for Amplifier No. 4 in mA
iHighLimit.4	Defines the high current limit for Amplifier No. 4 in mA

parameter	*description*
iDeltaLimit.4	Defines the delta current limit for Amplifier No. 4 in mA

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.iLowLimit.1	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iHighLimit.1	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iDeltaLimit.1	INTEGER	SETUP	IntegerRange (10 .. 1500)
config.iLowLimit.2	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iHighLimit.2	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iDeltaLimit.2	INTEGER	SETUP	IntegerRange (10 .. 1500)
config.iLowLimit.3	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iHighLimit.3	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iDeltaLimit.3	INTEGER	SETUP	IntegerRange (10 .. 1500)
config.iLowLimit.4	INTEGER	SETUP	IntegerRange (0 .. 1500)

name	*type*	*flags*	*range*
config.iHighLimit.4	INTEGER	SETUP	IntegerRange (0 .. 1500)
config.iDeltaLimit.4	INTEGER	SETUP	IntegerRange (10 .. 1500)
config.voltage.1	CHOICE	SETUP	EnumRange (0V 12V 15V 20V)
config.voltage.2	CHOICE	SETUP	EnumRange (0V 12V 15V 20V)
config.voltage.3	CHOICE	SETUP	EnumRange (0V 12V 15V 20V)
config.voltage.4	CHOICE	SETUP	EnumRange (0V 12V 15V 20V)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.model	TEXT		StringRange
internal.dummy	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Amplifier 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier Delta 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amplifier Delta 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amplifier Delta 3)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amplifier Delta 4)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU 1 24V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU 2 24V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU 1 6V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU 2 6V)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Internal VDC Control)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.269 Geosync-RCU-R2R

This page describes the device driver and the device window for the Geosync 1:2 Redundant switchover system series: **R2R-S50/S50**

It was tested with:

- 1:2 BDC Controller
- Model: R2R-S50/S50
- SW: 1.07 (2012-27-13) **Device window pages**

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The monitoring parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel as ASCII-character (e.g. @ for 64dec).

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
cableComp.1	Defines the cable compensation for unit 1.
cableComp.2	Defines the cable compensation for unit 2.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.cableComp.1	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.cableComp.2	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.model	TEXT		StringRange

name	*type*	*flags*	*range*
internal.dummy	TEXT		StringRange
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
priority.1	CHOICE		EnumRange (HIGH LOW)
priority.2	CHOICE		EnumRange (HIGH LOW)
tx.attn.0	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 0.0)
if.attn.0	FLOAT		DoubleRange (0.0 .. 0.0)
if.attn.1	FLOAT		DoubleRange (0.0 .. 0.0)
if.attn.2	FLOAT		DoubleRange (0.0 .. 0.0)
slope.0	FLOAT		DoubleRange (0.0 .. 0.0)
slope.1	FLOAT		DoubleRange (0.0 .. 0.0)
slope.2	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply A)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply B)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Converter Comms)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Converter Contacts)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.270 Geosync-RCU

This page describes the device driver and the device window for the Geosync LNA/LNB

redundancy controller

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The monitoring parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Remarks

- tcp port: 10001, configure via http://[ip]/secure/ltx_conf.htm
- default ip: see front panel display
- default address: @

Variables defined by this device driver

name	***type***	***flags***	***range***

9.271 Geosync-TLT

This page describes the device driver and the device window for the GeoSync Test Loop Translator.

The driver was tested with the following SW versions:

- R1.02 23/06/2017

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters.
- --- The meter readings.
- 1:N --- The redundancy parameters.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address. The default address is @ .
Serial	Remember to set the line parameters for the serial interface.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Remote Control

- Default TCP Port = 10001
- Default Address = @

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.loFreq	FLOAT		DoubleRange (0.000 .. 0.000)
info.inputLowFreq	FLOAT		DoubleRange (0.000 .. 0.000)
info.inputHighFreq	FLOAT		DoubleRange (0.000 .. 0.000)
info.outputLowFreq	FLOAT		DoubleRange (0.0 .. 0.0)
info.outputHighFreq	FLOAT		DoubleRange (0.0 .. 0.0)
info.numberOfBands	INTEGER		IntegerRange (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT	NOCOMPARE	DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT	NOCOMPARE	DoubleRange (0.0 .. 30.0)
externalRef	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL)
freqAdjust	INTEGER		IntegerRange (0 .. 4095)
config.bandSelect	INTEGER	SETUP	IntegerRange (1 .. 1)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ctrlMode	CHOICE	R/O	EnumRange R/O (LOCAL REMOTE)
internalOn	CHOICE	R/O	EnumRange R/O (ON OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Local oscillator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Test fault)

9.272 Gude-ExpertPowerControl

This page describes the device driver and the device window for the Gude Expert Power Control IP Switches and was tested with : Expert Power Control NET 2x6 (12x).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows and commands port status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- retries --- The number of communication retries which are made at application level.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
config.pulseTime.01	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.02	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.03	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.04	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.05	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.06	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.07	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.08	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.09	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.10	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.11	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.pulseTime.12	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupMode.01	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.02	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)

name	*type*	*flags*	*range*
config.startupMode.03	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.04	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.05	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.06	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.07	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.08	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.09	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.10	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.11	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupMode.12	CHOICE	SETUP	EnumRange DIS (OFF ON LAST STATE)
config.startupDelay.01	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.02	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.03	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.04	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.05	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.06	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.07	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.08	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.09	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.10	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
config.startupDelay.11	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)

name	*type*	*flags*	*range*
config.startupDelay.12	INTEGER	SETUP	IntegerRange DIS (0 .. 36000)
name.01	TEXT	R/O	StringRange R/O DIS
name.02	TEXT	R/O	StringRange R/O DIS
name.03	TEXT	R/O	StringRange R/O DIS
name.04	TEXT	R/O	StringRange R/O DIS
name.05	TEXT	R/O	StringRange R/O DIS
name.06	TEXT	R/O	StringRange R/O DIS
name.07	TEXT	R/O	StringRange R/O DIS
name.08	TEXT	R/O	StringRange R/O DIS
name.09	TEXT	R/O	StringRange R/O DIS
name.10	TEXT	R/O	StringRange R/O DIS
name.11	TEXT	R/O	StringRange R/O DIS
name.12	TEXT	R/O	StringRange R/O DIS
state.01	CHOICE		EnumRange DIS (ON OFF)
state.02	CHOICE		EnumRange DIS (ON OFF)
state.03	CHOICE		EnumRange DIS (ON OFF)
state.04	CHOICE		EnumRange DIS (ON OFF)
state.05	CHOICE		EnumRange DIS (ON OFF)
state.06	CHOICE		EnumRange DIS (ON OFF)
state.07	CHOICE		EnumRange DIS (ON OFF)
state.08	CHOICE		EnumRange DIS (ON OFF)
state.09	CHOICE		EnumRange DIS (ON OFF)
state.10	CHOICE		EnumRange DIS (ON OFF)
state.11	CHOICE		EnumRange DIS (ON OFF)
state.12	CHOICE		EnumRange DIS (ON OFF)
pulse.01	TEXT	NOPRESET	StringRange DIS

name	*type*	*flags*	*range*
pulse.02	TEXT	NOPRESET	StringRange DIS
pulse.03	TEXT	NOPRESET	StringRange DIS
pulse.04	TEXT	NOPRESET	StringRange DIS
pulse.05	TEXT	NOPRESET	StringRange DIS
pulse.06	TEXT	NOPRESET	StringRange DIS
pulse.07	TEXT	NOPRESET	StringRange DIS
pulse.08	TEXT	NOPRESET	StringRange DIS
pulse.09	TEXT	NOPRESET	StringRange DIS
pulse.10	TEXT	NOPRESET	StringRange DIS
pulse.11	TEXT	NOPRESET	StringRange DIS
pulse.12	TEXT	NOPRESET	StringRange DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Model not supported)

9.273 GundD-CompactCenter

This page describes the device driver and the device window for a Guntermann and Drunk Compact Center x2 KVM Switch. The driver is intended for monitoring only purpose as it only gives basic unit information regarding PSU, ethernet and error state of the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Overview
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
error.code	INTEGER	R/O	IntegerRange R/O (0 .. 0)
error.message	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (ETH 1 Down)
faults.04	ALARM	R/O	AlarmFlagRange R/O (ETH 2 Down)
faults.05	ALARM	R/O	AlarmFlagRange R/O (General Error)

9.274 HP-Powermeter-Legacy

This page describes the device driver and the device window for a single channel Hewlett-Packard Powermeter using SCPI Hewlett-Packard Interface Bus (GPIB/HP-IB) with the legacy classic protocol.

It was tested with:

- HP-436A

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the power reading and contains the controllable parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	Enter the GPIB address of the powermeter here (0..31)
config.displayMode	Select the desired power unit (W/dB(Rel)/db(Ref)/dBm).

parameter	*description*
config.calFactor	En/Disable usage of front panels cal factor.

Uncalibrated fault

The device driver shows all measurement faults reported by the power meter in one common fault flag called "Uncalibrated". This includes beside the calibration fault reported by the device also measurement over-/underflow faults and a pending power up selftest fault condition.

GPIB Interface setup

This device is operated through a GPIB (HPIB) interface. The sat-nms software actually supports two types of GPIB interfaces: The ENET100 by National Instruments and the Prologix Ethernet to GPIB interface. As the first one is no longer available, new installations should use the Prologix interface.

When configuring the sat-nms software for GPIB devices, type and parameters of the GPIB interface are set as the interface name, "GPIB" is selected as communication protocol. With a Prologix interface, the interface name is something like "PRLX:10.0.0.1", where PRLX selects the Prologix and 10.0.0.1 is an example for the IP address of the interface. With a ENET100 interface (supported for compatibility reasons), the interface name is "ENET:0" for the first interface, "ENET:1" for the second one.

Multiple devices can be operated a the same GPIB, these devices have to be configured all at the same interface. Configuring two or more GPIB interfaces with the same index ot IP address will cause communication faults and misconfigured devices.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.powermeterReply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.displayMode	CHOICE	SETUP	EnumRange (Watt dB(Rel) dB(Ref) dBm)
config.calFactor	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
power	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
range	CHOICE		EnumRange (1 2 3 4 5 AUTO)
zero	TEXT	NOPRESET	StringRange
state.autoZero	BOOLEAN	R/O	BooleanRange R/O (true false)
state.range	CHOICE	R/O	EnumRange R/O (1 2 3 4 5)
initalize	TEXT	R/O	StringRange R/O
internal.mantissa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.exponent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Invalid Measurement)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Under Range)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Over Range)

9.275 HP-Powermeter-SCPI

This page describes the device driver and the device window for the*HP-Powermeter-SCPI* device. This device reads the power reading from the device. It permits to set a user defined offset to the reading and to select the measurement range the device shall use (only applicable with some measurement sensors).

The driver has been tested with the Agilent E4418B powermeter, however it should be usable with all HP/Agilent/Keysight powermeters which can be controlled over the GPIB using the SCPI language. With double sensors powermeters the driver only controls the first (upper) measurement sensor. Switching the power measurement head to a fixed range (HIGH/LOW) is not supported my all types of sensors. If a sensor is used which does not permit to set the measurement range, the setting will fall back to AUTO if aou try to select LOW or HIGH.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the power reading and contains the controllable parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Enter the GPIB address of the powermeter here (0..31)

Uncalibrated fault

The device driver shows all measurement faults reported by the power meter in one common fault flag called "Uncalibrated". This includes beside the calibration fault reported by the device also measurement over-/underflow faults and a pending power up selftest fault condition.

GPIB Interface setup

This device is operated through a GPIB (HPIB) interface. The sat-nms software actually supports two types of GPIB interfaces: The ENET100 by National Instruments and the Prologix Ethernet to GPIB interface. As the first one is no longer available, new installations should use the Prologix interface.

When configuring the sat-nms software for GPIB devices, type and parameters of the GPIB interface are set as the interface name, "GPIB" is selected as communication protocol. With a Prologix interface, the interface name is something like "PRLX:10.0.0.1", where PRLX selects the Prologix and 10.0.0.1 is an example for the IP address of the interface. With a ENET100 interface (supported for compatibility reasons), the interface name is "ENET:0" for the first interface, "ENET:1" for the second one.

Multiple devices can be operated a the same GPIB, these devices have to be configured all at the same interface. Configuring two or more GPIB interfaces with the same index ot IP address will cause communication faults and misconfigured devices.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.manufacturer	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
power	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
range	CHOICE		EnumRange (AUTO LOW HIGH)
offset	FLOAT		DoubleRange (0.000 .. 0.000)
zero	TEXT	NOPRESET	StringRange
config.calFactor	FLOAT	SETUP	DoubleRange (1 .. 150)
internal.r	TEXT	R/O	StringRange R/O
internal.q	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Uncalibrated)

9.276 HP-Synthesizer-SCPI

This page describes the device driver and the device window for the*HP-Synthesizer-SCPI* device. This device permits to control the frequency, power level and output on/off settings of a HP signal synthesizer. It leaves all other parameters (e.g. modulation) untouched. These may be set at the device's font panel as required.

The driver has been tested with a HP 8648C synthesizer, however it should be usable with all HP/Agilent/Keysight synthesizers which can be controlled over the GPIB using the SCPI language.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Contains the controllable parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Enter the GPIB address of the powermeter here (0..31)

Uncalibrated fault

The device driver shows all faults reported by the synthesizer in one common fault flag called "Uncalibrated". This includes beside the calibration fault reported by the device also output power over/underload faults, the power up temperature warning and a pending power up selftest fault condition.

GPIB Interface setup

This device is operated through a GPIB (HPIB) interface. The sat-nms software actually supports two types of GPIB interfaces: The ENET100 by National Instruments and the Prologix Ethernet to GPIB interface. As the first one is no longer available, new installations should use the Prologix interface.

When configuring the sat-nms software for GPIB devices, type and parameters of the GPIB interface are set as the interface name, "GPIB" is selected as communication protocol. With a Prologix interface, the interface name is something like "PRLX:10.0.0.1", where PRLX selects the Prologix and 10.0.0.1 is an example for the IP address of the interface. With a ENET100 interface (supported for compatibility reasons), the interface name is "ENET:0" for the first interface, "ENET:1" for the second one.

Multiple devices can be operated a the same GPIB, these devices have to be configured all at the same interface. Configuring two or more GPIB interfaces with the same index ot IP address will cause communication faults and misconfigured devices.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.manufacturer	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
power	FLOAT		DoubleRange (0.00 .. 0.00)
rfOn	CHOICE		EnumRange (OFF ON)
internal.q	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Uncalibrated)

9.277 HWGroup-Poseidon2

This page describes the device driver for the Poseidon2 series SNMP based monitoring sensor and I/O controller from HWGroup.
Please note: To initialize the SNMP communication it is necessary to add a read and write community on the WebUI of the Poseidon2.

The usage of virtual outputs is not intended to be used with this driver.

The driver was tested with the following model:

- Poseidon2 3468 with 4 Inputs and 2 Outputs

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The I/O controller parameters and sensor measurements
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- sensors --- Defines the number of connected sensors between 0 and 16 of the device
- inputs --- Defines the number of available inputs (4 or 12) of the device
- outputs --- Defines the number of available outputs (0,2 or 4) of the device

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mac	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.sensors	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
config.inputs	CHOICE	SAVE SETUP	EnumRange (4 12)
config.outputs	CHOICE	SAVE SETUP	EnumRange (0 2 4)
internal.getUnit	CHOICE	R/O	EnumRange R/O (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
internal.setUnit	TEXT	R/O	StringRange R/O
sens.01.name	TEXT	R/O	StringRange R/O DIS
sens.02.name	TEXT	R/O	StringRange R/O DIS
sens.03.name	TEXT	R/O	StringRange R/O DIS
sens.04.name	TEXT	R/O	StringRange R/O DIS
sens.05.name	TEXT	R/O	StringRange R/O DIS
sens.06.name	TEXT	R/O	StringRange R/O DIS
sens.07.name	TEXT	R/O	StringRange R/O DIS
sens.08.name	TEXT	R/O	StringRange R/O DIS
sens.09.name	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
sens.10.name	TEXT	R/O	StringRange R/O DIS
sens.11.name	TEXT	R/O	StringRange R/O DIS
sens.12.name	TEXT	R/O	StringRange R/O DIS
sens.13.name	TEXT	R/O	StringRange R/O DIS
sens.14.name	TEXT	R/O	StringRange R/O DIS
sens.15.name	TEXT	R/O	StringRange R/O DIS
sens.16.name	TEXT	R/O	StringRange R/O DIS
sens.01.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.02.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.03.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.04.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.05.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.06.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)

name	*type*	*flags*	*range*
sens.07.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.08.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.09.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.10.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.11.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.12.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.13.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.14.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.15.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)

name	*type*	*flags*	*range*
sens.16.unit	CHOICE	R/O	EnumRange R/O DIS (CELSIUS FAHRENHEIT KELVIN PERCENT VOLT mA NONE PULSE SWITCH DEWPOINT ABSOLUTE HUMIDITY PRESSURE UNIVERSAL)
sens.01.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.02.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.03.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.04.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.05.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.06.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.07.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.08.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.09.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.10.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.11.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.12.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.13.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.14.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.15.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
sens.16.meas	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.01.state	CHOICE		EnumRange (ON OFF)
output.02.state	CHOICE		EnumRange (ON OFF)
output.03.state	CHOICE		EnumRange (ON OFF)
output.04.state	CHOICE		EnumRange (ON OFF)
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.01.type	CHOICE	R/O	EnumRange R/O (ON/OFF RTS DTR)
output.02.type	CHOICE	R/O	EnumRange R/O (ON/OFF RTS DTR)
output.03.type	CHOICE	R/O	EnumRange R/O (ON/OFF RTS DTR)
output.04.type	CHOICE	R/O	EnumRange R/O (ON/OFF RTS DTR)
output.01.mode	CHOICE		EnumRange (MANUAL ALARM TRIGGER EQUAL TRIGGER HIGH TRIGGER LOW)
output.02.mode	CHOICE		EnumRange (MANUAL ALARM TRIGGER EQUAL TRIGGER HIGH TRIGGER LOW)
output.03.mode	CHOICE		EnumRange (MANUAL ALARM TRIGGER EQUAL TRIGGER HIGH TRIGGER LOW)
output.04.mode	CHOICE		EnumRange (MANUAL ALARM TRIGGER EQUAL TRIGGER HIGH TRIGGER LOW)
input.01.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.02.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.03.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.04.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.05.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.06.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.07.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.08.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.09.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.10.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.11.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.12.state	CHOICE	R/O	EnumRange R/O (ON OFF)
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange

name	*type*	*flags*	*range*
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.01.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.02.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.03.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.04.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.05.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.06.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.07.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.08.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.09.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.10.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
input.11.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)

name	*type*	*flags*	*range*
input.12.mode	CHOICE		EnumRange (INACTIVE ACTIVE OFF ACTIVE ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Sensor 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sensor 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sensor 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Sensor 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Sensor 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sensor 6)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Sensor 7)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sensor 8)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Sensor 9)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Sensor 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sensor 11)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sensor 12)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sensor 13)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sensor 14)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Sensor 15)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Sensor 16)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input 9)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

9.278 HWGroup-STE

This page describes the device driver and the device window for the for the HWGroup STE Temperature/Humidity Sensor using SNMP.

It was tested with:

- Model STE firmware 2.1.7, 1x temperature and 1x humidity sensor
- Model STE firmware 2.2.3, 2x temperature sensors

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The measurement values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write

community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- sensors --- Defines the number of installed sensors on the unit.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.macAddress	TEXT	R/O	StringRange R/O
info.serialSens.1	TEXT	R/O	StringRange R/O
info.serialSens.2	TEXT	R/O	StringRange R/O
info.unit.1	CHOICE		EnumRange (NONE CELSIUS FAHRENHEIT KELVIN PERCENT)
info.unit.2	CHOICE		EnumRange (NONE CELSIUS FAHRENHEIT KELVIN PERCENT)
info.sysDescription	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysContact	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.sensors	CHOICE	SAVE SETUP	EnumRange (0 1 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
name.1	TEXT	R/O	StringRange R/O DIS
name.2	TEXT	R/O	StringRange R/O DIS
state.1	CHOICE	R/O	EnumRange R/O (INVALID NORMAL OUT-OF-RANGE-LOW OUT-OF-RANGE-HIGH ALARM-LOW ALARM-HIGH)
state.2	CHOICE	R/O	EnumRange R/O (INVALID NORMAL OUT-OF-RANGE-LOW OUT-OF-RANGE-HIGH ALARM-LOW ALARM-HIGH)
internal.value	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 invalid)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 low alarm)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 high alarm)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 out of range)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 invalid)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 low alarm)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 high alarm)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 out of range)

9.279 HWGroup-STE2

This page describes the device driver for the temperature measurement unit STE2. It allows displaying the measurements in kelvin, centigrade and fahrenheit.
Please note as the SNMP interface doesn't report the selected unit correctly make sure that the desired unit is only set by the MNC not from the WebUI.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Measurement value.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller as well as the model type.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
- hiThreshold --- Defines the high temperature threshold triggering the high temperature alarm.
- loThreshold --- Defines the low temperature threshold triggering the low temperature alarm.

- unit --- Defines the desired temperature unit for measurement and threshold display.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.loThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.unit	CHOICE	SAVE SETUP	EnumRange (NONE CELSIUS FAHRENHEIT KELVIN PERCENT)
internal.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
type	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low temp threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High temp threshold)

9.280 Haivision-Makito-X-Decoder

This page describes the device driver and the device window for the Haivison Makito-X IP/SRT Quad Decoder device. It supports RTP/UDP and SRT inputs.

The driver handles each decoder as a single device. For each additional decoder another device is needed. Desired Stream inputs can be selected via setup page.

Additional the used decoding instance can be configured/selected via setup choice**decoder**

The driver was tested with the following sw releases:

- **1.4.0-104**

Known issue: with the tested version there are no valid ip/ts input statistics when using UDP protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the IP input parameter settings and the operation control.
- --- Contains video output parameters and statistics from the TS stream as well as some general video settings.
- --- Contains audio output parameters from the TS stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- streamName --- Defines the selected output for this device from the list of available outputs.
- decoder --- Defines the used decoder for this device from the predefined list.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.sysError	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.streamName	CHOICE	SETUP	EnumRange (Unknown)
config.decoder	CHOICE	SAVE SETUP	EnumRange (DECODER-0 DECODER-1 DECODER-2 DECODER-3)
config.deviceComment	TEXT	SAVE SETUP	StringRange
streamId	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
decoderId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
videoId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.type	TEXT	R/O	StringRange R/O
internal.srtMode	TEXT	R/O	StringRange R/O
internal.ipMode	TEXT	R/O	StringRange R/O
internal.address	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
pause	TEXT	NOPRESET	StringRange
clear	TEXT	NOPRESET	StringRange
ipin.enable	CHOICE	R/O	EnumRange R/O (START PAUSE STOP)
ipin.type	CHOICE		EnumRange (SRT RTP UDP N/A)
ipin.mode	CHOICE		EnumRange (MULTICAST UNICAST DNS)
ipin.interface	TEXT	R/O	StringRange R/O
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (0 .. 65535)
ipin.fecEnable	CHOICE		EnumRange (OFF PROMPEG)
srtin.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtin.encryption	CHOICE		EnumRange (OFF ON)
srtin.key	TEXT	NOPRESET NOCOMPARE	StringRange
srtin.latency	INTEGER		IntegerRange (0 .. 0)
srtin.stream	TEXT		StringRange
ipin.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtin.srtVersion	TEXT	R/O	StringRange R/O
srtin.srtPeerVersion	TEXT	R/O	StringRange R/O
srtin.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.reconnects	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.localPort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.remotePort	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.maxBw	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtin.linkBw	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtin.packetsAck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.packetsNak	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.rtt	TEXT	R/O	StringRange R/O
srtin.buffer	TEXT	R/O	StringRange R/O
srtin.maxLatency	TEXT	R/O	StringRange R/O
video.frameRate	CHOICE		EnumRange (AUTO 85fps 75fps 60fps 59.94fps 50fps 30fps 29.97fps 25fps 24fps 23.98fps)
video.syncMode	CHOICE		EnumRange (PASSTHROUGH STC FIXED)
video.buffer	CHOICE		EnumRange (ADAPTIVE FIXED AUTO MULTISYNC)
video.delay	INTEGER		IntegerRange (0 .. 2000)
video.multiDelay	INTEGER		IntegerRange (0 .. 10000)
video.fail	CHOICE		EnumRange (FREEZE BLACK BLUE COLORBARS MUTE)

name	*type*	*flags*	*range*
video.failDelay	INTEGER		IntegerRange (0 .. 1000)
video.sdiMode	CHOICE		EnumRange (auto 1920x1080p 1920x1080i 1280x720p 720x576p 720x576i 720x480p 720x480i 1920x1200p 1680x1050p 1600x1200p 1600x900p 1440x900p 1400x1050p 1366x768p 1360x768p 1280x1024p 1280x800p 1280x768 1152x864p 1024x768p 800x600p 640x480)
video.dynamicRange	CHOICE		EnumRange (SDR AUTO HLG PQ UNKNOWN)
video.dropFrames	CHOICE		EnumRange (ON OFF)
video.codec	CHOICE	R/O	EnumRange R/O (INVALID H264 HEVC MPEG2 MPEG4)
video.format	TEXT	R/O	StringRange R/O
video.formatOut	TEXT	R/O	StringRange R/O
video.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
state.video	TEXT	R/O	StringRange R/O
audio.1.mode	CHOICE	R/O	EnumRange R/O (STEREO MONO)
audio.2.mode	CHOICE	R/O	EnumRange R/O (STEREO MONO)
audio.3.mode	CHOICE	R/O	EnumRange R/O (STEREO MONO)
audio.4.mode	CHOICE	R/O	EnumRange R/O (STEREO MONO)
audio.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.coding	TEXT	R/O	StringRange R/O
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.audio	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (TS)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)

9.281 Haivision-Makito-X

This page describes the device driver and the device window for the Haivison Makito-X IP/SRT Quad Encoder device. It supports RTP/UDP and SRT outputs.

The driver handles each stream output as a single device. For each additional output another device is needed. Desired Stream outputs have to be selected via setup page at first setup of a device using: *streamName* . Additional the used encoding instance can be configured/selected via setup *encoderName*

The driver was tested with the following sw releases:

- **3.0-37**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the MPEG output parameter settings and the operation control.
- --- Contains RTP/UDP and SRT output parameters and statistics.
- --- Contains video in/output parameters from the MPEG stream.
- --- Contains the parameters concerning the audio in/outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- streamName --- Defines the selected output for this device from the list of available outputs.
- encoderName --- Defines the used encoder for this device from the list of available outputs.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.streamName	CHOICE	SAVE SETUP	EnumRange (Unknown)
config.encoderName	CHOICE	SAVE SETUP	EnumRange (Unknown)
config.initAddress	TEXT	SAVE SETUP	StringRange
config.deviceComment	TEXT	SAVE SETUP	StringRange
streamId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
encoderId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
videoId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
noAudio	INTEGER	R/O	IntegerRange R/O (0 .. 8)
audio1Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio2Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio3Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio4Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio5Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)

name	*type*	*flags*	*range*
audio6Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio7Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
audio8Id	CHOICE	R/O	EnumRange R/O (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
internal.type	TEXT	R/O	StringRange R/O
internal.address	TEXT	R/O	StringRange R/O
internal.tos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.shaping	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.idleCells	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.delayedAudio	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.profile	TEXT	R/O	StringRange R/O
internal.codec	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
pause	TEXT	NOPRESET	StringRange
resume	TEXT	NOPRESET	StringRange
clear	TEXT	NOPRESET	StringRange
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
output.enable	CHOICE	R/O	EnumRange R/O (START PAUSE STOP)
output.type	CHOICE		EnumRange (N/A SRT RTP UDP)
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)

name	*type*	*flags*	*range*
output.program	INTEGER		IntegerRange (1 .. 65535)
output.service	TEXT		StringRange
ipout.interface	CHOICE		EnumRange (AUTO ETH-0 ETH-1)
ipout.addressType	CHOICE		EnumRange (IP DNS NONE)
ipout.address	TEXT		StringRange
ipout.port	INTEGER		IntegerRange (0 .. 65535)
ipout.tos	HEX		HexRange (0 .. 0)
ipout.mtu	INTEGER		IntegerRange (0 .. 0)
ipout.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.timingShaping	CHOICE		EnumRange (VBR CVBR CBR)
ipout.fecEnable	CHOICE		EnumRange (OFF PROMPEG VIDEO FURNANCE)
ipout.fecMode	CHOICE		EnumRange (A B)
ipout.fecColumns	INTEGER		IntegerRange (1 .. 20)
ipout.fecRows	INTEGER		IntegerRange (4 .. 20)
srtout.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
srtout.key	TEXT	NOPRESET	StringRange
srtout.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.stream	TEXT		StringRange
srtout.latency	INTEGER		IntegerRange (0 .. 0)
ipout.state	TEXT	R/O	StringRange R/O
ipout.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srtout.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.measLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
video.format	CHOICE		EnumRange (AUTO 1920x1080p 1920x1080i 1440x1080p 1440x1080i 960x1080p 960x1080i 1280x720p 720x576p 720x576i 540x576p 540x576i 352x576p 352x576i 352x288p 352x288i 448x336p)
video.profile	CHOICE		EnumRange (BASELINE MAIN HIGH HIGH-10 HIGH-422 MAIN-10 MAIN-10-422)
video.frameRate	CHOICE		EnumRange (AUTO 60fps 59fps 58fps 57fps 56fps 55fps 54fps 53fps 52fps 51fps 50fps 49fps 48fps 47fps 46fps 45fps 44fps 43fps 42fps 41fps 40fps 39fps 38fps 37fps 36fps 35fps 34fps 33fps 32fps 31fps 30fps 29fps 28fps 27fps 26fps 25fps 24fps 23fps 22fps 21fps 20fps 19fps 18fps 17fps 16fps 15fps 14fps 13fps 12fps 11fps 10fps 9fps 8fps 7fps 6fps 5fps 4fps 3fps 2fps 1fps)
video.resize	CHOICE		EnumRange (SCALE CROP)
video.rateControl	CHOICE		EnumRange (CBR VBR)
video.codec	CHOICE		EnumRange (H264 HEVC)
video.chroma	CHOICE		EnumRange (4:2:0-8Bit 4:2:0-10Bit 4:2:2-8Bit 4:2:2-10Bit)
video.aspectRatio	CHOICE		EnumRange (AUTO WSS-AFD 4:3 16:9 3:2 5:3 5:4 16:10 17:9)

name	*type*	*flags*	*range*
video.gopLength	INTEGER		IntegerRange (1 .. 1000)
video.gopStruct	CHOICE		EnumRange (IP I IBP IBBP IBBBP IBBBBP)
video.datarate	FLOAT		DoubleRange (0.000 .. 0.000)
video.maxrate	FLOAT		DoubleRange (0.000 .. 0.000)
video.sdiMode	CHOICE		EnumRange (AUTO 1920x1080p 1920x1080i 1440x1080p 1440x1080i 960x1080p 960x1080i 1280x720p 720x576p 720x576i 540x576p 540x576i 352x576p 352x576i 352x288p 352x288i 448x336p)
video.timecode	CHOICE		EnumRange (NONE VITC SYSTEM)
video.countMode	CHOICE		EnumRange (UTC-CONVERSION SMPTE12M-1)
video.sdiFormat	TEXT	R/O	StringRange R/O
video.pid	INTEGER		IntegerRange (0 .. 8190)
audio.1.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.1.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.1.samplingRate	CHOICE		EnumRange (48000Hz)

name	*type*	*flags*	*range*
audio.1.datarate	INTEGER		IntegerRange (32 .. 448)
audio.1.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.1.language	TEXT		StringRange
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.1.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)
audio.2.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.2.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.2.samplingRate	CHOICE		EnumRange (48000Hz)
audio.2.datarate	INTEGER		IntegerRange (32 .. 448)
audio.2.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.2.language	TEXT		StringRange
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)

name	*type*	*flags*	*range*
audio.2.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.2.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)
audio.3.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.3.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.3.samplingRate	CHOICE		EnumRange (48000Hz)
audio.3.datarate	INTEGER		IntegerRange (32 .. 448)
audio.3.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.3.language	TEXT		StringRange
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.3.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)

name	*type*	*flags*	*range*
audio.4.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.4.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.4.samplingRate	CHOICE		EnumRange (48000Hz)
audio.4.datarate	INTEGER		IntegerRange (32 .. 448)
audio.4.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.4.language	TEXT		StringRange
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.4.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)

name	*type*	*flags*	*range*
audio.5.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.5.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.5.samplingRate	CHOICE		EnumRange (48000Hz)
audio.5.datarate	INTEGER		IntegerRange (32 .. 448)
audio.5.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.5.language	TEXT		StringRange
audio.5.pid	INTEGER		IntegerRange (0 .. 8191)
audio.5.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.5.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)

name	*type*	*flags*	*range*
audio.6.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.6.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.6.samplingRate	CHOICE		EnumRange (48000Hz)
audio.6.datarate	INTEGER		IntegerRange (32 .. 448)
audio.6.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.6.language	TEXT		StringRange
audio.6.pid	INTEGER		IntegerRange (0 .. 8191)
audio.6.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.6.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)

name	*type*	*flags*	*range*
audio.7.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.7.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.7.samplingRate	CHOICE		EnumRange (48000Hz)
audio.7.datarate	INTEGER		IntegerRange (32 .. 448)
audio.7.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.7.language	TEXT		StringRange
audio.7.pid	INTEGER		IntegerRange (0 .. 8191)
audio.7.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.7.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)

name	*type*	*flags*	*range*
audio.8.enable	CHOICE		EnumRange (OFF Audio Encoder 0 Audio Encoder 1 Audio Encoder 2 Audio Encoder 3 Audio Encoder 4 Audio Encoder 5 Audio Encoder 6 Audio Encoder 7 Audio Encoder 8 Audio Encoder 9 Audio Encoder 10 Audio Encoder 11 Audio Encoder 12 Audio Encoder 13 Audio Encoder 14 Audio Encoder 15 Audio Encoder 16 Audio Encoder 17 Audio Encoder 18 Audio Encoder 19 Audio Encoder 20 Audio Encoder 21 Audio Encoder 22 Audio Encoder 23 Audio Encoder 24 Audio Encoder 25 Audio Encoder 26 Audio Encoder 27 Audio Encoder 28 Audio Encoder 29 Audio Encoder 30 Audio Encoder 31)
audio.8.mode	CHOICE		EnumRange (STEREO MONO MONO-RIGHT)
audio.8.samplingRate	CHOICE		EnumRange (48000Hz)
audio.8.datarate	INTEGER		IntegerRange (32 .. 448)
audio.8.coding	CHOICE		EnumRange (RAW MPEG2-ADTS MPEG4-ADTS MPEG4-LOAS)
audio.8.language	TEXT		StringRange
audio.8.pid	INTEGER		IntegerRange (0 .. 8191)
audio.8.source	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.8.embedded	CHOICE		EnumRange (1/2 3/4 5/6 7/8 9/10 11/12 13/14 15/16)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)

9.282 Haivision-SRT-Gateway

This page describes the device driver and the device window for the Haivision SRT Gateway, controlled via REST HTTPs.

It was developed and tested with SW release**5.5.230220.1702** It supports one IP input which

can be SRT,UDP,RTP,RTMP,RTSP. When using SRT a redundant in and output can be defined. It supports up to 4 IP outputs from the following formats: UDP,RTP,SRT,HTTP (HLS) The units base configuration needs to be done in advance.

The driver can only handle one route / gateway instance. As The Haivision can handle more than one "gateway". Each instance has to be setup in a single device. The respective instance has to be configured at first setup via config.route. The list of available routes/gateways is determined at first driver start. If new routes are added the device driver has to be set Out-Of-Service and back to Operation to fetch those changes.

Please note. As the number of simultaneous users accessing the Gateway is limited it is necessary to share the login information and token through the instances. Therefore we recommend that only the first unit contains the management ip address username and password in the device setup. All other units can use the same token when defining it´s address as **@DeviceName** whereby DeviceName is the device name of the unit containing the ip and login credentials in its device setup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains all input parameters
- --- Contains IP output 1 settings and statistics
- --- Contains IP output 2 settings and statistics
- --- Contains IP output 3 settings and statistics
- --- Contains IP output 4 settings and statistics
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Haivision Gateway.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added. When using more than one device/route, the address can refer to existing device via @DeviceName
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

parameter	*description*
route	Defines the desired route/gateway instance.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.load	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.name	TEXT	R/O	StringRange R/O
info.route	TEXT	R/O	StringRange R/O
config.route	CHOICE	SAVE SETUP	EnumRange (?)
config.readBackDelay	INTEGER	SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
opState	CHOICE		EnumRange (STARTED STOPPED)
restart	TEXT	NOPRESET	StringRange
ipin.alias	TEXT		StringRange
ipin.1.interface	CHOICE		EnumRange (AUTO)
ipin.1.port	INTEGER		IntegerRange (1 .. 65535)
ipin.1.address	TEXT		StringRange
ipin.protocol	CHOICE		EnumRange (SRT UDP RTP RTMP RTSP)
ipin.mode	CHOICE		EnumRange (MULTICAST UNICAST CALLER LISTENER RENDEZVOUS N/A)
ipin.2.interface	CHOICE		EnumRange (AUTO)
ipin.2.port	INTEGER		IntegerRange (1 .. 65535)
ipin.2.address	TEXT		StringRange
ipin.fecEnable	CHOICE		EnumRange (ON OFF)
ipin.retainHeader	CHOICE		EnumRange (ON OFF)
ipin.sourceAddress	TEXT		StringRange
srtin.latency	INTEGER		IntegerRange (20 .. 8000)
srtin.buffer	INTEGER		IntegerRange (1 .. 1000000)
srtin.encryption	CHOICE		EnumRange (AUTO AES-GCM)
srtin.key	TEXT		StringRange

name	*type*	*flags*	*range*
srtin.redundancy	CHOICE		EnumRange (NONE ACTIVE-ACTIVE ACTIVE-BACKUP ANY)
srtin.stream	TEXT		StringRange
srtin.fecEnable	CHOICE		EnumRange (OFF ON)
srtin.fecColumns	INTEGER		IntegerRange (0 .. 0)
srtin.fecRows	INTEGER		IntegerRange (0 .. 0)
srtin.fecArq	CHOICE		EnumRange (NEVER ON-REQUEST)
srtin.fecLayout	CHOICE		EnumRange (STAIRCASE EVEN)
rtmpin.stream	TEXT		StringRange
rtmpin.mode	CHOICE		EnumRange (CONSUMER PUBLISHER)
rtspin.username	TEXT		StringRange
rtspin.password	TEXT		StringRange
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.elapsed	TEXT	R/O	StringRange R/O
ipin.state	TEXT	R/O	StringRange R/O
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.usedBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.signalLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsUnreco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.packetslost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.retransRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srtin.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.negLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.lossRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.packetsSkip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtin.decryption	TEXT	R/O	StringRange R/O
srtin.bufferMeas	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.alias	TEXT		StringRange
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP SRT HTTP)
ipout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS MULTICAST UNICAST HLS)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (1 .. 65535)
ipout.1.interface	CHOICE		EnumRange (AUTO)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.mtu	INTEGER		IntegerRange (280 .. 1500)
ipout.1.tos	INTEGER		IntegerRange (0 .. 255)
ipout.1.retainHeader	CHOICE		EnumRange (OFF ON)
ipout.1.fecEnable	CHOICE		EnumRange (OFF VF ON)
ipout.1.fecLevel	CHOICE		EnumRange (A B)
ipout.1.fecBlock	CHOICE		EnumRange (OFF ON)
ipout.1.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.1.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.1.shaping	CHOICE		EnumRange (OFF ON)
ipout.1.maxBitrate	FLOAT		DoubleRange (0.000 .. 0.000)
srtout.1.latency	INTEGER		IntegerRange (20 .. 8000)

name	*type*	*flags*	*range*
srtout.1.overhead	INTEGER		IntegerRange (5 .. 100)
srtout.1.encryption	CHOICE		EnumRange (OFF AES128 AES256)
srtout.1.stream	TEXT		StringRange
srtout.1.redundancy	CHOICE		EnumRange (NONE ACTIVE-ACTIVE ACTIVE-BACKUP ANY)
srtout.1.key	TEXT		StringRange
srtout.1.limit	INTEGER		IntegerRange (0 .. 0)
srtout.1.fecEnable	CHOICE		EnumRange (OFF ON)
srtout.1.fecArq	CHOICE		EnumRange (NEVER ON-REQUEST)
srtout.1.fecLayout	CHOICE		EnumRange (STAIRCASE EVEN)
srtout.1.fecColumns	INTEGER		IntegerRange (0 .. 0)
srtout.1.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.1.address.2	TEXT		StringRange
ipout.1.port.2	INTEGER		IntegerRange (1 .. 65535)
ipout.1.interface.2	CHOICE		EnumRange (AUTO)
ipout.1.ttl.2	INTEGER		IntegerRange (1 .. 255)
ipout.1.mtu.2	INTEGER		IntegerRange (280 .. 1500)
ipout.1.tos.2	INTEGER		IntegerRange (0 .. 255)
hlsout.1.url	TEXT	R/O	StringRange R/O
hlsout.1.stream	TEXT	R/O	StringRange R/O
hlsout.1.encryption	CHOICE		EnumRange (ON OFF)
hlsout.1.multitrack	CHOICE		EnumRange (ON OFF)
hlsout.1.segments	INTEGER		IntegerRange (1 .. 1000)
hlsout.1.duration	FLOAT		DoubleRange (1.00 .. 2000000.00)
ipout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.elapsed	TEXT	R/O	StringRange R/O
ipout.1.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipout.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.1.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.1.usedBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.1.signalLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsUnreco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.retransRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.negLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.lossRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsSkip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.decryption	TEXT	R/O	StringRange R/O
srtout.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.curBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.1.maxBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.1.estBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.1.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.alias	TEXT		StringRange
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP SRT HTTP)

name	*type*	*flags*	*range*
ipout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS MULTICAST UNICAST HLS)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (1 .. 65535)
ipout.2.interface	CHOICE		EnumRange (AUTO)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.mtu	INTEGER		IntegerRange (280 .. 1500)
ipout.2.tos	INTEGER		IntegerRange (0 .. 255)
ipout.2.retainHeader	CHOICE		EnumRange (OFF ON)
ipout.2.fecEnable	CHOICE		EnumRange (OFF VF ON)
ipout.2.fecLevel	CHOICE		EnumRange (A B)
ipout.2.fecBlock	CHOICE		EnumRange (OFF ON)
ipout.2.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.2.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.2.shaping	CHOICE		EnumRange (OFF ON)
ipout.2.maxBitrate	FLOAT		DoubleRange (0.000 .. 0.000)
srtout.2.latency	INTEGER		IntegerRange (20 .. 8000)
srtout.2.overhead	INTEGER		IntegerRange (5 .. 100)
srtout.2.encryption	CHOICE		EnumRange (OFF AES128 AES256)
srtout.2.stream	TEXT		StringRange
srtout.2.redundancy	CHOICE		EnumRange (NONE ACTIVE-ACTIVE ACTIVE-BACKUP ANY)
srtout.2.key	TEXT		StringRange
srtout.2.limit	INTEGER		IntegerRange (0 .. 0)
srtout.2.fecEnable	CHOICE		EnumRange (OFF ON)
srtout.2.fecArq	CHOICE		EnumRange (NEVER ON-REQUEST)
srtout.2.fecLayout	CHOICE		EnumRange (STAIRCASE EVEN)

name	*type*	*flags*	*range*
srtout.2.fecColumns	INTEGER		IntegerRange (0 .. 0)
srtout.2.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.2.address.2	TEXT		StringRange
ipout.2.port.2	INTEGER		IntegerRange (1 .. 65535)
ipout.2.interface.2	CHOICE		EnumRange (AUTO)
ipout.2.ttl.2	INTEGER		IntegerRange (1 .. 255)
ipout.2.mtu.2	INTEGER		IntegerRange (280 .. 1500)
ipout.2.tos.2	INTEGER		IntegerRange (0 .. 255)
hlsout.2.url	TEXT	R/O	StringRange R/O
hlsout.2.stream	TEXT	R/O	StringRange R/O
hlsout.2.encryption	CHOICE		EnumRange (ON OFF)
hlsout.2.multitrack	CHOICE		EnumRange (ON OFF)
hlsout.2.segments	INTEGER		IntegerRange (1 .. 1000)
hlsout.2.duration	FLOAT		DoubleRange (1.00 .. 2000000.00)
ipout.2.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.elapsed	TEXT	R/O	StringRange R/O
ipout.2.state	TEXT	R/O	StringRange R/O
ipout.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.2.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.2.usedBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.2.signalLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.packetsUnreco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
srtout.2.retransRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.negLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.lossRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsSkip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.decryption	TEXT	R/O	StringRange R/O
srtout.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.2.curBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.2.maxBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.2.estBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.2.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.alias	TEXT		StringRange
ipout.3.enable	CHOICE		EnumRange (OFF ON)
ipout.3.protocol	CHOICE		EnumRange (UDP RTP SRT HTTP)
ipout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS MULTICAST UNICAST HLS)
ipout.3.address	TEXT		StringRange
ipout.3.port	INTEGER		IntegerRange (1 .. 65535)
ipout.3.interface	CHOICE		EnumRange (AUTO)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.3.mtu	INTEGER		IntegerRange (280 .. 1500)
ipout.3.tos	INTEGER		IntegerRange (0 .. 255)
ipout.3.retainHeader	CHOICE		EnumRange (OFF ON)
ipout.3.fecEnable	CHOICE		EnumRange (OFF VF ON)
ipout.3.fecLevel	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
ipout.3.fecBlock	CHOICE		EnumRange (OFF ON)
ipout.3.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.3.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.3.shaping	CHOICE		EnumRange (OFF ON)
ipout.3.maxBitrate	FLOAT		DoubleRange (0.000 .. 0.000)
srtout.3.latency	INTEGER		IntegerRange (20 .. 8000)
srtout.3.overhead	INTEGER		IntegerRange (5 .. 100)
srtout.3.encryption	CHOICE		EnumRange (OFF AES128 AES256)
srtout.3.stream	TEXT		StringRange
srtout.3.redundancy	CHOICE		EnumRange (NONE ACTIVE-ACTIVE ACTIVE-BACKUP ANY)
srtout.3.key	TEXT		StringRange
srtout.3.limit	INTEGER		IntegerRange (0 .. 0)
srtout.3.fecEnable	CHOICE		EnumRange (OFF ON)
srtout.3.fecArq	CHOICE		EnumRange (NEVER ON-REQUEST)
srtout.3.fecLayout	CHOICE		EnumRange (STAIRCASE EVEN)
srtout.3.fecColumns	INTEGER		IntegerRange (0 .. 0)
srtout.3.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.3.address.2	TEXT		StringRange
ipout.3.port.2	INTEGER		IntegerRange (1 .. 65535)
ipout.3.interface.2	CHOICE		EnumRange (AUTO)
ipout.3.ttl.2	INTEGER		IntegerRange (1 .. 255)
ipout.3.mtu.2	INTEGER		IntegerRange (280 .. 1500)
ipout.3.tos.2	INTEGER		IntegerRange (0 .. 255)
hlsout.3.url	TEXT	R/O	StringRange R/O
hlsout.3.stream	TEXT	R/O	StringRange R/O
hlsout.3.encryption	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
hlsout.3.multitrack	CHOICE		EnumRange (ON OFF)
hlsout.3.segments	INTEGER		IntegerRange (1 .. 1000)
hlsout.3.duration	FLOAT		DoubleRange (1.00 .. 2000000.00)
ipout.3.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.elapsed	TEXT	R/O	StringRange R/O
ipout.3.state	TEXT	R/O	StringRange R/O
ipout.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.3.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.3.usedBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.3.signalLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.packetsUnreco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.3.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.retransRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.negLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.lossRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsSkip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.decryption	TEXT	R/O	StringRange R/O
srtout.3.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.3.curBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.3.maxBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.3.estBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
srtout.3.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.4.alias	TEXT		StringRange
ipout.4.enable	CHOICE		EnumRange (OFF ON)
ipout.4.protocol	CHOICE		EnumRange (UDP RTP SRT HTTP)
ipout.4.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS MULTICAST UNICAST HLS)
ipout.4.address	TEXT		StringRange
ipout.4.port	INTEGER		IntegerRange (1 .. 65535)
ipout.4.interface	CHOICE		EnumRange (AUTO)
ipout.4.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.4.mtu	INTEGER		IntegerRange (280 .. 1500)
ipout.4.tos	INTEGER		IntegerRange (0 .. 255)
ipout.4.retainHeader	CHOICE		EnumRange (OFF ON)
ipout.4.fecEnable	CHOICE		EnumRange (OFF VF ON)
ipout.4.fecLevel	CHOICE		EnumRange (A B)
ipout.4.fecBlock	CHOICE		EnumRange (OFF ON)
ipout.4.fecColumns	INTEGER		IntegerRange (4 .. 20)
ipout.4.fecRows	INTEGER		IntegerRange (4 .. 20)
ipout.4.shaping	CHOICE		EnumRange (OFF ON)
ipout.4.maxBitrate	FLOAT		DoubleRange (0.000 .. 0.000)
srtout.4.latency	INTEGER		IntegerRange (20 .. 8000)
srtout.4.overhead	INTEGER		IntegerRange (5 .. 100)
srtout.4.encryption	CHOICE		EnumRange (OFF AES128 AES256)
srtout.4.stream	TEXT		StringRange
srtout.4.redundancy	CHOICE		EnumRange (NONE ACTIVE-ACTIVE ACTIVE-BACKUP ANY)

name	*type*	*flags*	*range*
srtout.4.key	TEXT		StringRange
srtout.4.limit	INTEGER		IntegerRange (0 .. 0)
srtout.4.fecEnable	CHOICE		EnumRange (OFF ON)
srtout.4.fecArq	CHOICE		EnumRange (NEVER ON-REQUEST)
srtout.4.fecLayout	CHOICE		EnumRange (STAIRCASE EVEN)
srtout.4.fecColumns	INTEGER		IntegerRange (0 .. 0)
srtout.4.fecRows	INTEGER		IntegerRange (0 .. 0)
ipout.4.address.2	TEXT		StringRange
ipout.4.port.2	INTEGER		IntegerRange (1 .. 65535)
ipout.4.interface.2	CHOICE		EnumRange (AUTO)
ipout.4.ttl.2	INTEGER		IntegerRange (1 .. 255)
ipout.4.mtu.2	INTEGER		IntegerRange (280 .. 1500)
ipout.4.tos.2	INTEGER		IntegerRange (0 .. 255)
hlsout.4.url	TEXT	R/O	StringRange R/O
hlsout.4.stream	TEXT	R/O	StringRange R/O
hlsout.4.encryption	CHOICE		EnumRange (ON OFF)
hlsout.4.multitrack	CHOICE		EnumRange (ON OFF)
hlsout.4.segments	INTEGER		IntegerRange (1 .. 1000)
hlsout.4.duration	FLOAT		DoubleRange (1.00 .. 2000000.00)
ipout.4.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.4.elapsed	TEXT	R/O	StringRange R/O
ipout.4.state	TEXT	R/O	StringRange R/O
ipout.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.4.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.4.usedBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.4.signalLosses	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.4.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipout.4.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.4.packetsUnreco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.4.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.retransRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.negLatency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.lossRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsSkip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.decryption	TEXT	R/O	StringRange R/O
srtout.4.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.4.curBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.4.maxBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.4.estBandwidth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
srtout.4.connections	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.list	TEXT	R/O	StringRange R/O
srtout.2.list	TEXT	R/O	StringRange R/O
srtout.3.list	TEXT	R/O	StringRange R/O
srtout.4.list	TEXT	R/O	StringRange R/O
srtout.1.details	CHOICE	NOPRESET	EnumRange (?)
srtout.2.details	CHOICE	NOPRESET	EnumRange (?)
srtout.3.details	CHOICE	NOPRESET	EnumRange (?)
srtout.4.details	CHOICE	NOPRESET	EnumRange (?)
internal.deviceId	TEXT	R/O	StringRange R/O
internal.routeId	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.sourceId	TEXT	R/O	StringRange R/O
internal.dest1Id	TEXT	R/O	StringRange R/O
internal.dest2Id	TEXT	R/O	StringRange R/O
internal.dest3Id	TEXT	R/O	StringRange R/O
internal.dest4Id	TEXT	R/O	StringRange R/O
internal.in	TEXT	R/O	StringRange R/O
internal.output1	TEXT	R/O	StringRange R/O
internal.output2	TEXT	R/O	StringRange R/O
internal.output3	TEXT	R/O	StringRange R/O
internal.output4	TEXT	R/O	StringRange R/O
internal.modeIn	TEXT	R/O	StringRange R/O
internal.modeOut1	TEXT	R/O	StringRange R/O
internal.modeOut2	TEXT	R/O	StringRange R/O
internal.modeOut3	TEXT	R/O	StringRange R/O
internal.modeOut4	TEXT	R/O	StringRange R/O
internal.redIn	TEXT	R/O	StringRange R/O
internal.redOut1	TEXT	R/O	StringRange R/O
internal.redOut2	TEXT	R/O	StringRange R/O
internal.redOut3	TEXT	R/O	StringRange R/O
internal.redOut4	TEXT	R/O	StringRange R/O
internal.out1	TEXT	R/O	StringRange R/O
internal.out2	TEXT	R/O	StringRange R/O
internal.out3	TEXT	R/O	StringRange R/O
internal.out4	TEXT	R/O	StringRange R/O
internal.ipinAddress	TEXT	R/O	StringRange R/O
internal.out1Address	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.out2Address	TEXT	R/O	StringRange R/O
internal.out3Address	TEXT	R/O	StringRange R/O
internal.out4Address	TEXT	R/O	StringRange R/O
internal.url1	TEXT	R/O	StringRange R/O
internal.url2	TEXT	R/O	StringRange R/O
internal.url3	TEXT	R/O	StringRange R/O
internal.url4	TEXT	R/O	StringRange R/O
internal.enoR	TEXT	R/O	StringRange R/O
internal.dest1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dest4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hasDest1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasDest2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasDest3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasDest4	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Device Summary Warn)

9.283 Harris-Panacea

This page describes the device driver and the device window for a Harris (formally Leitch) ASI Switch Matrix (Panacea series) with upto 16 inputs and outputs. The driver supports only interfaces on Level 0.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Harris Matrix do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .

For network configuration use telnet to set IP address, netmask and gateway first (with terminal protocol, the command *show menu e* shows all ethernet related commands. Then configure the driver interface in the sat-nms software to IP-Address:23.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.username	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.password	TEXT	SAVE SETUP	StringRange
info.swversion	TEXT	R/O	StringRange R/O
info.bootloader	TEXT	R/O	StringRange R/O
info.fpga	TEXT	R/O	StringRange R/O
info.matrix	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.id.frame	TEXT	R/O	StringRange R/O
info.id.cntr	TEXT	R/O	StringRange R/O
info.id.license	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.01	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.02	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.03	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)

name	*type*	*flags*	*range*
output.04	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.05	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.06	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.07	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.08	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.09	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)

name	*type*	*flags*	*range*
output.10	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.11	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.12	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.13	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.14	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.15	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)

name	*type*	*flags*	*range*
output.16	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.17	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.18	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.19	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.20	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.21	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)

name	*type*	*flags*	*range*
output.22	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.23	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.24	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.25	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.26	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.27	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)

name	*type*	*flags*	*range*
output.28	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.29	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.30	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.31	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
output.32	CHOICE		EnumRange DIS (00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015 00016 00017 00018 00019 00020 00021 00022 00023 00024 00025 00026 00027 00028 00029 00030 00031 00032)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
reset	TEXT		StringRange
internal.manual	TEXT	R/O	StringRange R/O
internal.board	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.284 Hilkom-BNova-SNMP

This page describes the device driver and the device window for the*Hilkom-BNova-SNMP* device. This device monitors the state of a Hilkom B-Nova head end receiver. The driver is monitoring only, it polls a (limited) number of status parameters from all modules installed in the B-Nova frame. The driver automatically detects which modules are installed.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Overview page, shows which modules are installed and if they are OK.
- 1 --- Detailled status page for each individual module
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.deviceName	TEXT	R/O	StringRange R/O
info.hwSerialNumber	TEXT	R/O	StringRange R/O
info.hwType	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
slot1.present	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.type	TEXT	R/O	StringRange R/O
slot1.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port1.type	TEXT	R/O	StringRange R/O
slot1.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot1.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot1.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot1.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port2.type	TEXT	R/O	StringRange R/O
slot1.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot1.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot1.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot1.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot1.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port3.type	TEXT	R/O	StringRange R/O
slot1.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot1.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot1.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot1.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port4.type	TEXT	R/O	StringRange R/O
slot1.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot1.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot1.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot2.present	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.type	TEXT	R/O	StringRange R/O
slot2.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port1.type	TEXT	R/O	StringRange R/O
slot2.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
slot2.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot2.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot2.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port2.type	TEXT	R/O	StringRange R/O
slot2.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot2.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot2.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot2.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port3.type	TEXT	R/O	StringRange R/O
slot2.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot2.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot2.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot2.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port4.type	TEXT	R/O	StringRange R/O
slot2.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot2.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot2.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot3.present	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot3.type	TEXT	R/O	StringRange R/O
slot3.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port1.type	TEXT	R/O	StringRange R/O
slot3.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot3.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot3.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot3.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port2.type	TEXT	R/O	StringRange R/O
slot3.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot3.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot3.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot3.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port3.type	TEXT	R/O	StringRange R/O
slot3.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot3.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot3.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot3.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port4.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot3.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot3.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot3.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot4.present	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.type	TEXT	R/O	StringRange R/O
slot4.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port1.type	TEXT	R/O	StringRange R/O
slot4.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot4.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot4.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot4.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port2.type	TEXT	R/O	StringRange R/O
slot4.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot4.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot4.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot4.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port3.type	TEXT	R/O	StringRange R/O
slot4.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
slot4.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot4.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot4.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port4.type	TEXT	R/O	StringRange R/O
slot4.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot4.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot4.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot5.present	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.type	TEXT	R/O	StringRange R/O
slot5.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port1.type	TEXT	R/O	StringRange R/O
slot5.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot5.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot5.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot5.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port2.type	TEXT	R/O	StringRange R/O
slot5.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot5.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)

name	*type*	*flags*	*range*
slot5.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot5.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port3.type	TEXT	R/O	StringRange R/O
slot5.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot5.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot5.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot5.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port4.type	TEXT	R/O	StringRange R/O
slot5.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot5.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot5.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot6.present	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.type	TEXT	R/O	StringRange R/O
slot6.port1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port1.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port1.type	TEXT	R/O	StringRange R/O
slot6.port1.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port1.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot6.port1.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot6.port1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
slot6.port2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port2.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port2.type	TEXT	R/O	StringRange R/O
slot6.port2.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port2.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot6.port2.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot6.port2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot6.port3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port3.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port3.type	TEXT	R/O	StringRange R/O
slot6.port3.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port3.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot6.port3.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot6.port3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slot6.port4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port4.output	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port4.type	TEXT	R/O	StringRange R/O
slot6.port4.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
slot6.port4.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
slot6.port4.ber	FLOAT	R/O	DoubleRange R/O (0.000E+00 .. 0.000E+00)
slot6.port4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 4)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 6)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Interface 1.1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Interface 1.2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Interface 1.3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Interface 1.4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Interface 2.1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Interface 2.2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Interface 2.3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Interface 2.4)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Interface 3.1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Interface 3.2)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Interface 3.3)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Interface 3.4)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Interface 4.1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Interface 4.2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Interface 4.3)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Interface 4.4)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Interface 5.1)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Interface 5.2)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Interface 5.3)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Interface 5.4)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Interface 6.1)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Interface 6.2)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Interface 6.3)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Interface 6.4)

9.285 Hiltron-ACU2

This page describes the device driver and the device window for the Hiltron ACU2 antenna controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Antenna Pointing.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter* --- *description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- azLoLimit --- Azimuth Software Low Limit
- azHiLimit --- Azimuth Software High Limit
- elLoLimit --- Elevation Software Low Limit
- elHiLimit --- Elevation Software High Limit
- plLoLimit --- Polarization Software Low Limit
- plHiLimit --- Polarization Software High Limit

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysUpTime	TEXT	R/O	StringRange R/O
info.beaconRx	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.hasTracking	CHOICE	SAVE SETUP	EnumRange (YES NO)

name	*type*	*flags*	*range*
config.hasPol	CHOICE	SAVE SETUP	EnumRange (YES NO)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azMotor	TEXT	R/O	StringRange R/O
state.elMotor	TEXT	R/O	StringRange R/O
state.plMotor	TEXT	R/O	StringRange R/O
state.onTarget.az	CHOICE	R/O	EnumRange R/O (NO YES)
state.onTarget.el	CHOICE	R/O	EnumRange R/O (NO YES)
state.onTarget.pl	CHOICE	R/O	EnumRange R/O (NO YES)
state.onTarget	BOOLEAN	R/O	BooleanRange R/O (true false)
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.moving.az	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.el	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.pl	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
motorStop	TEXT	NOPRESET	StringRange
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O
trackMode	CHOICE		EnumRange (OFF TRACKING OPTIMISE SEARCH)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
target.name	TEXT	R/O	StringRange R/O
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 199)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 199)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ hi hard limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ lo hard limit)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ hi soft limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ lo soft limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EL hi hard limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (EL lo hard limit)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL hi soft limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL lo soft limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PL hi hard limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PL lo hard limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL hi soft limit)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PL lo soft limit)
faults.21	ALARM	R/O	AlarmFlagRange R/O (AZ Emergency stop)
faults.22	ALARM	R/O	AlarmFlagRange R/O (AZ Drive Voltage)
faults.23	ALARM	R/O	AlarmFlagRange R/O (EL Emergency stop)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EL Drive Voltage)
faults.25	ALARM	R/O	AlarmFlagRange R/O (PL Emergency stop)
faults.26	ALARM	R/O	AlarmFlagRange R/O (PL Drive Voltage)
faults.31	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.32	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.34	ALARM	R/O	AlarmFlagRange R/O (AZ FU communication)

name	*type*	*flags*	*range*
faults.35	ALARM	R/O	AlarmFlagRange R/O (EL FU communication)
faults.36	ALARM	R/O	AlarmFlagRange R/O (PL FU communication)
faults.37	ALARM	R/O	AlarmFlagRange R/O (AZ communication)
faults.38	ALARM	R/O	AlarmFlagRange R/O (EL communication)
faults.39	ALARM	R/O	AlarmFlagRange R/O (PL communication)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Beacon Communication)
faults.42	ALARM	R/O	AlarmFlagRange R/O (TLE data age)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Beacon signal)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Steptrack fail)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Position invalid)
faults.46	ALARM	R/O	AlarmFlagRange R/O (No TLE)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Time NTP)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Step Size)
faults.49	ALARM	R/O	AlarmFlagRange R/O (General Tracking)

9.286 Hiltron-Deice

This page describes the device driver and the device window for the Hiltron De-Icing controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Antenna Pointing.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- tempSensors --- Number of installed temperature sensors
- currentSensors --- Number of installed current sensors

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysUpTime	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.tempSensors	CHOICE	SAVE SETUP	EnumRange (0 1 2)
config.currentSensors	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5)
deice.mode	CHOICE		EnumRange (AUTO MANUAL-ON MANUAL-OFF)
deice.state	TEXT	R/O	StringRange R/O
deice.fallback	INTEGER	R/O	IntegerRange R/O (0 .. 0)
deice.snowSensor	TEXT	R/O	StringRange R/O
temperature.1	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temperature.1.label	TEXT	R/O	StringRange R/O DIS
temperature.2.label	TEXT	R/O	StringRange R/O DIS
temperature.1.request	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
temperature.2.request	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
current.1	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.3	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.4	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.5	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.1.label	TEXT	R/O	StringRange R/O DIS
current.2.label	TEXT	R/O	StringRange R/O DIS
current.3.label	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
current.4.label	TEXT	R/O	StringRange R/O DIS
current.5.label	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Snow sensor)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Heater Comm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temp. sensor 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temp. sensor 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Current limit 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Current limit 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Current limit 3)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Current limit 4)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Current limit 5)

9.287 Hiltron-HCS-Frame

This page describes the device driver to check the state of an Hiltron HCS3 main frame / chassis. This driver does not include any modules. Use the matching Hiltron-HCS-* driver instead.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification

consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysUpTime	TEXT	R/O	StringRange R/O

9.288 Hiltron-HCS-PS

This page describes the device driver and the device window for the*Hiltron-HCS-PS* device. This device monitors the state of a power supply module within a HCS3 frame. To manage multiple modules of this kind in one HCS3 controller you have to configure multiple instances of the driver, all using the same IP address.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- The module number (1..14) to address in the HCS3 controller.

Remarks

If you change the 'module' configuration parameter, the module identification at the info page is not updated automatically. This happens with the next restart of the software. To force an update of these readings, set the device OUT-OF-SERVICE and then OPERATIONAL again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.slot	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18)
extVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
intVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
intCurrent	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fault	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (low ext. voltage)
faults.02	ALARM	R/O	AlarmFlagRange R/O (high ext. voltage)
faults.03	ALARM	R/O	AlarmFlagRange R/O (low int. voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (high int. voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (low int. current)
faults.06	ALARM	R/O	AlarmFlagRange R/O (high int. current)
faults.07	ALARM	R/O	AlarmFlagRange R/O (low temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (high temperature)

9.289 Hiltron-HCS-Power

This page describes the device driver and the device window for the*Hiltron-HCS-Power* device.

This device controls the LNB or Optical Link Power. To manage multiple modules of this kind in one HCS3 controller you have to configure multiple instances of the driver, all using the same IP address.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- The module number (1..14) to address in the HCS3 controller.

Remarks

If you change the 'module' configuration parameter, the module identification at the info page is not updated automatically. This happens with the next restart of the software. To force an update of these readings, set the device OUT-OF-SERVICE and then OPERATIONAL again.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.slot	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14)
voltage	FLOAT		DoubleRange (0.0 .. 0.0)
current	FLOAT		DoubleRange (0.00 .. 0.00)
state	CHOICE	R/O	EnumRange R/O (OK Voltage Fail Current Fail Voltage+Current Fail Shortcut Shortcut+Voltage Fail Shortcut+Current Fail Shortcut+Voltage+Current Fail)
internal.fault	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Voltage)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Current)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Shortcut)

9.290 Hiltron-HCS-SSPA

This page describes the device driver and the device window for the*Hiltron-HCS-SSPA* device. This device monitors the state a SSPA connected via HCS3 frame. To manage multiple modules of this kind in one HCS3 controller you have to configure multiple instances of the driver, all using the same IP address.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters.
- --- Shows the actual state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- The module number (1..14) to address in the HCS3 controller.

Remarks

If you change the 'module' configuration parameter, the module identification at the info page is not updated automatically. This happens with the next restart of the software. To force an update of these readings, set the device OUT-OF-SERVICE and then OPERATIONAL again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.slot	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18)
tx.on	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state	TEXT	R/O	StringRange R/O
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary fail)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RF Alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF FAIL)
faults.09	ALARM	R/O	AlarmFlagRange R/O (BUC alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (BUC fail)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temperature alarm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature fail)
faults.13	ALARM	R/O	AlarmFlagRange R/O (High reflected RF)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Intern reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (HPA Communication)

9.291 Hiltron-HCS-Switch

This page describes the device driver and the device window for the*Hiltron-HCS-Switch* device. This device controls one waveguide switch / redundancy switch module in a Hiltron HCS3 controller. To manage multiple switches in one HCS3 controller you have to configure multiple instances of the driver, all using the same IP address.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets / shows the actual switch state.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- The module number (1..14) to address in the HCS3 controller.

Remarks

- The 'position' variable knows beside the switch positions 'A' and 'B' the states 'NONE' and 'FAIL'. These are for state display only, these are not commandable. The device ignores position commanding it to 'NONE' or 'FAIL'.
- If you change the 'module' configuration parameter, the module identification at the info page is not updated automatically. This happens with the next restart of the software. To force an update of these readings, set the device OUT-OF-SERVICE and then OPERATIONAL again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.slot	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14)
position	CHOICE		EnumRange (NONE A B FAIL)
redundancyMode	CHOICE		EnumRange (MANUAL AUTOMATIC)
posA.strength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
posA.state	CHOICE	R/O	EnumRange R/O (OK FAULT)
posB.strength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
posB.state	CHOICE	R/O	EnumRange R/O (OK FAULT)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Indifferent position)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch failure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal comms)

9.292 Hiltron-HCS-TWTA

This page describes the device driver and the device window for the*Hiltron-HCS-TWTA* device. This device monitors the state a TWTA connected via HCS3 frame. To manage multiple modules of this kind in one HCS3 controller you have to configure multiple instances of the driver, all using the same IP address.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The operational parameters.
- --- Shows the actual state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- The module number (1..14) to address in the HCS3 controller.

Remarks

If you change the 'module' configuration parameter, the module identification at the info page is not updated automatically. This happens with the next restart of the software. To force an update of these readings, set the device OUT-OF-SERVICE and then OPERATIONAL again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.slot	TEXT	R/O	StringRange R/O
info.card	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (ON OFF)
tx.beam	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state	TEXT	R/O	StringRange R/O
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary fail)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RF Alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF FAIL)
faults.09	ALARM	R/O	AlarmFlagRange R/O (BUC alarm)
faults.10	ALARM	R/O	AlarmFlagRange R/O (BUC fail)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temperature alarm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature fail)
faults.13	ALARM	R/O	AlarmFlagRange R/O (High reflected RF)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Intern reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (HPA Communication)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.293 Hiltron-Multi-Deice

This page describes the device driver and the device window for the Hiltron Multi De-Icing controller HCS-4 with more than one DCU.

The desired DCU has to be configured via device setup page on first setup. For each DCU a separate instance of this driver has to configured.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Antenna Pointing.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- Position of installed DCU module in HCS-4 between 1 to 7.

- tempSensors --- Number of installed temperature sensors
- currentSensors --- Number of installed current sensors
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysUpTime	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
internal.id1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.id2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.id3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.id4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.id5	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.module	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7)
config.tempSensors	CHOICE	SAVE SETUP	EnumRange (0 1 2)
config.currentSensors	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5)
config.deviceComment	TEXT	R/O	StringRange R/O
deice.mode	CHOICE		EnumRange (AUTO MANUAL-ON MANUAL-OFF)
deice.state	TEXT	R/O	StringRange R/O
deice.fallback	INTEGER	R/O	IntegerRange R/O (0 .. 0)
deice.snowSensor	TEXT	R/O	StringRange R/O
temperature.1	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temperature.1.label	TEXT	R/O	StringRange R/O DIS
temperature.2.label	TEXT	R/O	StringRange R/O DIS
temperature.1.request	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
temperature.2.request	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
current.1	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.3	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.4	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.5	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
current.1.label	TEXT	R/O	StringRange R/O DIS
current.2.label	TEXT	R/O	StringRange R/O DIS
current.3.label	TEXT	R/O	StringRange R/O DIS
current.4.label	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
current.5.label	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Snow sensor)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Heater Comm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temp. sensor 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temp. sensor 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Current limit 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Current limit 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Current limit 3)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Current limit 4)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Current limit 5)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Module Communication)

9.294 Hitec-AFP-ACU

This page describes the monitoring only device driver and the device window for the Hitec Antenna Front Panel. The driver covers the ACU related parameters

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Driver Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (General Summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Hardware Summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Limit)

9.295 Hitec-AFP-Deice

This page describes the monitoring only device driver and the device window for the Hitec Antenna Front Panel. The driver covers the De-icing related parameters.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
deice.status	CHOICE		EnumRange (OFF AUTOMATIC)
deice.auto	CHOICE		EnumRange (OFF ON)
deice.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
feedblower.status	CHOICE	R/O	EnumRange R/O (OFF ON)
state.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
feedblower.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.a	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
deice.b	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.c	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.subref	BOOLEAN	R/O	BooleanRange R/O (true false)
temp.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.aped	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.driveCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.fanDriveCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.waterTank	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.aped	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HVAC Summary)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Deicing Summary)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Feed blower fuse)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Feed blower)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Deicing fuse)
faults.13	ALARM	R/O	AlarmFlagRange R/O (High Subref.Temp.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Deicing trapech)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
internal.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.296 Hitec-HACU-01-NAOS

This page describes the device driver and the device window for the*Hitec HACH-01-NAOS* antenna controller. This driver version is customized for NAOS project. It is based on HACU-01_AcuInterfaceControlDocument_V3.13 from 20220518.

The driver uses the Hitec-HACU protocol.
IP address and port should be given by the interface in the device setup.
Default communication port is 12340

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- ---
- --- Satellite Pass and TLE Parameters
- --- Monopulse tracking parameters (present if configured as monopulse ACU)
- --- Monopulse calibration parameters (present if configured as monopulse ACU)
- --- Steptrack parameters
- --- Adaptive tracking parameters
- --- External Switches
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
smallJogDelta	The (small) amount to move the antenna in jog mode.
largeJogDelta	The (large) amount to move the antenna in jog mode.
beaconInTargets	Selects wether to include the beacon frequency in target memories.
operatorPasswd	The operator password, required to communicate with the ACU.
model	Selects if the ACU is a step track or a monopulse model.

parameter	*description*
coordinateSystem	Selects the coordinates systems between Antenna Axis and Geographical
aliasChannelA	Allows to define an alias name used for channel A
aliasChannelB	Allows to define an alias name used for channel B
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.hasAzimuth	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasElevation	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPolarization	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasMPA	BOOLEAN	R/O	BooleanRange R/O (true false)
info.time	TEXT	R/O	StringRange R/O
config.smallJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)

name	*type*	*flags*	*range*
config.largeJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.beaconInTargets	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.model	CHOICE	SAVE SETUP	EnumRange (STEPTRACK MONOPULSE)
config.coordinateSystem	CHOICE	SETUP	EnumRange (ANTENNA AXIS GEOGRAPHICAL)
config.aliasChannelA	TEXT	SAVE SETUP	StringRange
config.aliasChannelB	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
initialized	TEXT	R/O	StringRange R/O
internal.once	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.h	HEX	R/O	HexRange R/O (0 .. 0)
internal.rffaults	HEX	R/O	HexRange R/O (0 .. 0)
internal.acufaults	HEX	R/O	HexRange R/O (0 .. 0)
internal.v1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
prev.mode.az	TEXT	R/O	StringRange R/O
prev.mode.el	TEXT	R/O	StringRange R/O
prev.mode.pl	TEXT	R/O	StringRange R/O
internal.login	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.xtrempos1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.xtrempos2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.xtrempos3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.submode1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.submode2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.trackmode	TEXT	R/O	StringRange R/O
internal.h1	HEX	R/O	HexRange R/O (0 .. 0)
internal.h2	HEX	R/O	HexRange R/O (0 .. 0)
internal.h3	HEX	R/O	HexRange R/O (0 .. 0)
internal.h4	HEX	R/O	HexRange R/O (0 .. 0)
internal.h5	HEX	R/O	HexRange R/O (0 .. 0)
internal.h6	HEX	R/O	HexRange R/O (0 .. 0)
internal.h7	HEX	R/O	HexRange R/O (0 .. 0)
internal.h8	HEX	R/O	HexRange R/O (0 .. 0)
internal.range	TEXT	R/O	StringRange R/O
internal.channel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.target	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tList	TEXT	R/O	StringRange R/O
internal.tSlot	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tName	TEXT	R/O	StringRange R/O
internal.tInit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.r	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.mpapos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mpa0	TEXT	R/O	StringRange R/O
internal.mpa1	TEXT	R/O	StringRange R/O
internal.mpa2	TEXT	R/O	StringRange R/O
internal.mpa3	TEXT	R/O	StringRange R/O
internal.mpa4	TEXT	R/O	StringRange R/O
internal.mpa1pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.mpa2pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.mpa3pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.mpa4pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
mpa	FLOAT		DoubleRange (0.000 .. 0.000)
mpa.name	TEXT	R/O	StringRange R/O
mpa.mode	CHOICE		EnumRange (MANUAL AUTOMATIC)
mpa.target	CHOICE		EnumRange (0 1 2 3 4)
mode.az	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)

name	*type*	*flags*	*range*
mode.el	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
mode.pl	CHOICE		EnumRange (UNUSED STAND-BY POINTING STOW UNSTOW MANUAL)
mode.mpa	CHOICE		EnumRange (UNUSED STAND-BY POINTING STOW UNSTOW MANUAL)
trackMode	CHOICE		EnumRange (INVALID STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
subMode	TEXT	R/O	StringRange R/O
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
target	INTEGER		IntegerRange (0 .. 100)
target.name	CHOICE		EnumRange (NONE)
target.frequency	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
target.nomLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
target.available	BOOLEAN	R/O	BooleanRange R/O (true false)
target.type	CHOICE	R/O	EnumRange R/O (UNDEFINED TLE I11 TABULAR STAR SUN OTHER)
target.accessible	BOOLEAN	R/O	BooleanRange R/O (true false)
onTarget	BOOLEAN	R/O	BooleanRange R/O (true false)
target.valid	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
target.timeElapsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tle.name	TEXT		StringRange
tle.line1	TEXT		StringRange
tle.line2	TEXT		StringRange
tle.save	TEXT	NOPRESET	StringRange
tle.load	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mpa	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.plVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.time	TEXT	R/O	StringRange R/O
pass.end.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.end.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.end.time	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
pass.maxEl.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.maxEl.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.maxEl.time	TEXT	R/O	StringRange R/O
pass.valid	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.beacon1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.beacon2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.delta1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.delta2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dtrLock	CHOICE	R/O	EnumRange R/O (UNLOCKED LOCKED)
channel	CHOICE		EnumRange (INVALID A B)
channel.mode	CHOICE		EnumRange (MANUAL AUTOMATIC)
frequency	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
freqBand	CHOICE		EnumRange (S-BAND X-BAND)
dtr.control	CHOICE		EnumRange (UNDEFINED NOT IN CONTROL IN CONTROL)
doppler.timestamp	TEXT	R/O	StringRange R/O
doppler.freqOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
doppler.velocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stop	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
jog	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
pointing	TEXT	NOPRESET	StringRange
monopulse	TEXT	NOPRESET	StringRange
monopulseCal	TEXT	NOPRESET	StringRange
steptrack	TEXT	NOPRESET	StringRange
adaptivetrack	TEXT	NOPRESET	StringRange
programtrack	TEXT	NOPRESET	StringRange
scan	TEXT	NOPRESET	StringRange
autotrack	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange
slew	TEXT	NOPRESET	StringRange
mp.modelType	CHOICE		EnumRange (WHITE-NOISE SINGER)
mp.processNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.maneuverTime	FLOAT		DoubleRange (0.000 .. 0.000)
mp.maxAzimuthSpeed	FLOAT		DoubleRange (0.000 .. 0.000)
mp.measurementNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.signalThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.errorThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.signalFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.distanceFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.modelResetThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.status	TEXT	R/O	StringRange R/O
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.dtrLock	TEXT	R/O	StringRange R/O
mpc.patternType	CHOICE		EnumRange DIS (CIRCULAR)
mpc.azReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.elReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.direction	CHOICE		EnumRange DIS (CCW CW)
mpc.patternStart	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.patternRadius	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfMeasurements	INTEGER		IntegerRange DIS (0 .. 0)
mpc.signalThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.startThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfSamples	INTEGER		IntegerRange DIS (0 .. 0)
mpc.stabilizeDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.measurementDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.xelevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.xelevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
mpc.elevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.status	TEXT	R/O	StringRange R/O DIS
mpc.xelevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.phaseError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.directionError	TEXT	R/O	StringRange R/O DIS
mpc.residual	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.phasingMode	CHOICE		EnumRange DIS (MANUAL AUTOMATIC)
mpc.minFreqSpacing	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
mpc.entryWarningTime	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.entryAlarmTime	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.delete	INTEGER		IntegerRange (0 .. 1)
mpc.getslots	INTEGER		IntegerRange (0 .. 1)
mpc.index.l	INTEGER		IntegerRange (-1 .. 100)
mpc.index.r	INTEGER		IntegerRange (-1 .. 100)
mpc.curpoll	INTEGER		IntegerRange (-1 .. 2)
mpc.curpolr	INTEGER		IntegerRange (-1 .. 2)
mpc.activepol	INTEGER		IntegerRange (-1 .. 2)
st.algorithm	CHOICE		EnumRange (BASIC GRADIENT)
st.maxIterations	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
st.corrThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.cycleTime	FLOAT		DoubleRange (0.000 .. 0.000)
st.beamWidth	FLOAT		DoubleRange (0.000 .. 0.000)
st.startThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.signalThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.optThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.sequence	CHOICE		EnumRange (CROSS-ELEV ELEVATION)
st.direction	CHOICE		EnumRange (DEC1-DEC2 DEC1-INC2 INC1-INC2 INC1-DEC2)
st.noOfSamples	INTEGER		IntegerRange (0 .. 0)
st.stepSize	FLOAT		DoubleRange (0.000 .. 0.000)
st.stabilizeDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.measurementDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.model	CHOICE		EnumRange (NO-MODEL MODEL-1 MODEL-2 MODEL-3)
st.decayRate0	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate1	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate2	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate3	FLOAT		DoubleRange (0.000 .. 0.000)
st.status	TEXT	R/O	StringRange R/O
st.xelevationOffs	FLOAT		DoubleRange (0.000 .. 0.000)
st.elevationOffs	FLOAT		DoubleRange (0.000 .. 0.000)
st.currentModel	TEXT	R/O	StringRange R/O
st.currentDistance	FLOAT		DoubleRange (0.000 .. 0.000)
at.samplingPeriod	FLOAT		DoubleRange (0.000 .. 0.000)
at.positionNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
at.speedNoiseSD	FLOAT		DoubleRange (0.00000 .. 0.00000)
at.deviationNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.okPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.warningPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.errorPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.enabled	CHOICE		EnumRange (DISABLED ENABLED)
at.status	TEXT	R/O	StringRange R/O
position.01	CHOICE		EnumRange (A B)
position.02	CHOICE		EnumRange (A B)
position.03	CHOICE		EnumRange (A B)
position.04	CHOICE		EnumRange (A B)
position.05	CHOICE		EnumRange (A B)
position.06	CHOICE		EnumRange (A B)
position.07	CHOICE		EnumRange (A B)
position.08	CHOICE		EnumRange (A B)
position.09	CHOICE		EnumRange (A B)
position.10	CHOICE		EnumRange (A B)
position.11	CHOICE		EnumRange (A B)
position.12	CHOICE		EnumRange (A B)
position.13	CHOICE		EnumRange (A B)
position.14	CHOICE		EnumRange (A B)
position.15	CHOICE		EnumRange (A B)
position.16	CHOICE		EnumRange (A B)
wgstat.1	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.2	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
wgstat.3	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.4	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.5	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.6	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.7	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.8	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.9	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.10	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.11	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.12	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.13	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.14	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.15	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.16	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.1	HEX		HexRange (0 .. 0)
wgcmd.2	HEX		HexRange (0 .. 0)
wgcmd.3	HEX		HexRange (0 .. 0)
wgcmd.4	HEX		HexRange (0 .. 0)
wgcmd.5	HEX		HexRange (0 .. 0)
wgcmd.6	HEX		HexRange (0 .. 0)
wgcmd.7	HEX		HexRange (0 .. 0)
wgcmd.8	HEX		HexRange (0 .. 0)
wgcmd.9	HEX		HexRange (0 .. 0)
wgcmd.10	HEX		HexRange (0 .. 0)
wgcmd.11	HEX		HexRange (0 .. 0)
wgcmd.12	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
wgcmd.13	HEX		HexRange (0 .. 0)
wgcmd.14	HEX		HexRange (0 .. 0)
wgcmd.15	HEX		HexRange (0 .. 0)
wgcmd.16	HEX		HexRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power distribution)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Motor 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Motor 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Az/El Encoder)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access door)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Limit switches 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Limit switches 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.11	ALARM	R/O	AlarmFlagRange R/O (General)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF switches)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ACU summary)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Axis summary 1)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Axis summary 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (SCU summary)
faults.21	ALARM	R/O	AlarmFlagRange R/O (RF Switch 01)
faults.22	ALARM	R/O	AlarmFlagRange R/O (RF Switch 02)
faults.23	ALARM	R/O	AlarmFlagRange R/O (RF Switch 03)
faults.24	ALARM	R/O	AlarmFlagRange R/O (RF Switch 04)
faults.25	ALARM	R/O	AlarmFlagRange R/O (RF Switch 05)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF Switch 06)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RF Switch 07)
faults.28	ALARM	R/O	AlarmFlagRange R/O (RF Switch 08)
faults.29	ALARM	R/O	AlarmFlagRange R/O (RF Switch 09)
faults.30	ALARM	R/O	AlarmFlagRange R/O (RF Switch 10)
faults.31	ALARM	R/O	AlarmFlagRange R/O (RF Switch 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (RF Switch 12)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RF Switch 13)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RF Switch 14)

name	*type*	*flags*	*range*
faults.35	ALARM	R/O	AlarmFlagRange R/O (RF Switch 15)
faults.36	ALARM	R/O	AlarmFlagRange R/O (RF Switch 16)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Dehydrator)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Tracking LNA)
faults.39	ALARM	R/O	AlarmFlagRange R/O (DNC controller)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Low Elevation)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Beacon Lost)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Beacon Saturated)
faults.43	ALARM	R/O	AlarmFlagRange R/O (DTR in Local)
faults.44	ALARM	R/O	AlarmFlagRange R/O (DTR Summary)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Monopulse Cal Needed)
faults.46	ALARM	R/O	AlarmFlagRange R/O (DTR Configuration)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Time Reference)
faults.48	ALARM	R/O	AlarmFlagRange R/O (SCU Comms Alarm 1)
faults.49	ALARM	R/O	AlarmFlagRange R/O (SCU Comms Alarm 2)
faults.50	ALARM	R/O	AlarmFlagRange R/O (IERS EOP EOT warn)

name	*type*	*flags*	*range*
faults.51	ALARM	R/O	AlarmFlagRange R/O (IERS EOP EOT alarm)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 1)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 2)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 3)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 4)
faults.56	ALARM	R/O	AlarmFlagRange R/O (DTR Comms Alarm 1)
faults.57	ALARM	R/O	AlarmFlagRange R/O (DTR Comms Alarm 2)
faults.58	ALARM	R/O	AlarmFlagRange R/O (End of Pred Model Valid)
faults.59	ALARM	R/O	AlarmFlagRange R/O (End of Orbit Model Valid)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Below Horizon Mask)
faults.61	ALARM	R/O	AlarmFlagRange R/O (HW Diagnostic Summary)

9.297 Hitec-HACU-01-V3.02

This page describes the device driver and the device window for the*Hitec HACH-01-V3.02* antenna controller. This is a legacy driver version intended for use with the Hitec HACU model 3000 (monopulse) and software version 2.5

The driver uses the Hitec-HACU protocol. IP address and port should be given by the interface in the device setup. Default communication port is 12340

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- ---
- --- Satellite Pass and TLE Parameters
- --- Monopulse tracking parameters (present if configured as monopulse ACU)
- --- Monopulse calibration parameters (present if configured as monopulse ACU)
- --- Steptrack parameters
- --- Adaptive tracking parameters
- --- External Switches
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
smallJogDelta	The (small) amount to move the antenna in jog mode.
largeJogDelta	The (large) amount to move the antenna in jog mode.
beaconInTargets	Selects wether to include the beacon frequency in target memories.
operatorPasswd	The operator password, required to communicate with the ACU.
model	Selects if the ACU is a step track or a monopulse model.
hasPolarization	Selects whether the ACU controls a polarization axis.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.smallJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.largeJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.beaconInTargets	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.model	CHOICE	SAVE SETUP	EnumRange (STEPTRACK MONOPULSE)
config.hasPolarization	CHOICE	SAVE SETUP	EnumRange (YES NO)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.h	HEX	R/O	HexRange R/O (0 .. 0)
internal.v1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
prev.mode.az	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
prev.mode.el	TEXT	R/O	StringRange R/O
prev.mode.pl	TEXT	R/O	StringRange R/O
internal.login	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
mode.az	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
mode.el	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
mode.pl	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
state.mode1	TEXT	R/O	StringRange R/O
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.delta1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.delta2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dtrLock	BOOLEAN	R/O	BooleanRange R/O (true false)
stop	TEXT	NOPRESET	StringRange
jog	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
pointing	TEXT	NOPRESET	StringRange
monopulse	TEXT	NOPRESET	StringRange
monopulseCal	TEXT	NOPRESET	StringRange
steptrack	TEXT	NOPRESET	StringRange
adaptivetrack	TEXT	NOPRESET	StringRange
programtrack	TEXT	NOPRESET	StringRange
scan	TEXT	NOPRESET	StringRange
autotrack	TEXT	NOPRESET	StringRange
mp.modelType	CHOICE		EnumRange (WHITE-NOISE SINGER)
mp.processNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.maneuverTime	FLOAT		DoubleRange (0.000 .. 0.000)
mp.maxAzimuthSpeed	FLOAT		DoubleRange (0.000 .. 0.000)
mp.measurementNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.signalThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.errorThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.signalFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.distanceFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.modelResetThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.status	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.dtrLock	TEXT	R/O	StringRange R/O
mpc.patternType	CHOICE		EnumRange DIS (CIRCULAR)
mpc.azReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.elReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.plReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.direction	CHOICE		EnumRange DIS (CCW CW)
mpc.patternStart	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.patternRadius	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfMeasurements	INTEGER		IntegerRange DIS (0 .. 0)
mpc.signalThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.startThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfSamples	INTEGER		IntegerRange DIS (0 .. 0)
mpc.stabilizeDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.measurementDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
mpc.xelevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.xelevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.status	TEXT	R/O	StringRange R/O DIS
mpc.xelevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.phaseError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.directionError	TEXT	R/O	StringRange R/O DIS
mpc.residual	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
st.algorithm	CHOICE		EnumRange (BASIC GRADIENT)
st.maxIterations	INTEGER		IntegerRange (0 .. 0)
st.corrThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.cycleTime	FLOAT		DoubleRange (0.000 .. 0.000)
st.beamWidth	FLOAT		DoubleRange (0.000 .. 0.000)
st.startThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.signalThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.optThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
st.sequence	CHOICE		EnumRange (CROSS-ELEV ELEVATION)

name	*type*	*flags*	*range*
st.direction	CHOICE		EnumRange (DEC1-DEC2 DEC1-INC2 INC1-INC2 INC1-DEC2)
st.noOfSamples	INTEGER		IntegerRange (0 .. 0)
st.stepSize	FLOAT		DoubleRange (0.000 .. 0.000)
st.stabilizeDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.measurementDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.model	CHOICE		EnumRange (NO-MODEL MODEL-1 MODEL-2 MODEL-3)
st.decayRate0	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate1	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate2	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate3	FLOAT		DoubleRange (0.000 .. 0.000)
st.status	TEXT	R/O	StringRange R/O
st.xelevationOffs	FLOAT		DoubleRange (0.000 .. 0.000)
st.elevationOffs	FLOAT		DoubleRange (0.000 .. 0.000)
st.currentModel	TEXT	R/O	StringRange R/O
st.currentDistance	FLOAT		DoubleRange (0.000 .. 0.000)
at.samplingPeriod	FLOAT		DoubleRange (0.000 .. 0.000)
at.positionNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.speedNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.deviationNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.okPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.warningPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.errorPeriodThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
at.enabled	CHOICE		EnumRange (DISABLED ENABLED)
at.status	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power distribution)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Motor 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Motor 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Az/El Encoder)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access door)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Limit switches 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Limit switches 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.11	ALARM	R/O	AlarmFlagRange R/O (General)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF switches)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ACU summary)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Axis summary 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Axis summary 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (SCU summary)

9.298 Hitec-HACU-01-V3

This page describes the device driver and the device window for the*Hitec HACH-01-V3* antenna controller. This driver has been tested with version 3.15 of the Hitec HACU ICD.

The driver uses the Hitec-HACU protocol. IP address and port should be given by the interface in the device setup. Default communication port is 12340

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- ---
- --- Satellite Pass and TLE Parameters
- --- Monopulse tracking parameters (present if configured as monopulse ACU)
- --- Monopulse calibration parameters (present if configured as monopulse ACU)
- --- Steptrack parameters
- --- External Switches
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
smallJogDelta	The (small) amount to move the antenna in jog mode.
largeJogDelta	The (large) amount to move the antenna in jog mode.
beaconInTargets	Selects wether to include the beacon frequency in target memories.
operatorPasswd	The operator password, required to communicate with the ACU.
coordinateSystem	Selects the coordinates systems between Antenna Axis and Geographical
aliasChannelA	Allows to define an alias name used for channel A
aliasChannelB	Allows to define an alias name used for channel B
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.hasAzimuth	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasElevation	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPolarization	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasMPA	BOOLEAN	R/O	BooleanRange R/O (true false)
info.time	TEXT	R/O	StringRange R/O
config.smallJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.largeJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.beaconInTargets	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.coordinateSystem	CHOICE	SETUP	EnumRange (ANTENNA AXIS GEOGRAPHICAL)

name	*type*	*flags*	*range*
config.aliasChannelA	TEXT	SAVE SETUP	StringRange
config.aliasChannelB	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
initialized	TEXT	R/O	StringRange R/O
internal.once	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.h	HEX	R/O	HexRange R/O (0 .. 0)
internal.rffaults	HEX	R/O	HexRange R/O (0 .. 0)
internal.acufaults	HEX	R/O	HexRange R/O (0 .. 0)
internal.v1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
prev.mode.az	TEXT	R/O	StringRange R/O
prev.mode.el	TEXT	R/O	StringRange R/O
prev.mode.pl	TEXT	R/O	StringRange R/O
internal.login	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.xtrempos1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.xtrempos2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.xtrempos3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.submode1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.submode2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.trackmode	TEXT	R/O	StringRange R/O
internal.h1	HEX	R/O	HexRange R/O (0 .. 0)
internal.h2	HEX	R/O	HexRange R/O (0 .. 0)
internal.h3	HEX	R/O	HexRange R/O (0 .. 0)
internal.h4	HEX	R/O	HexRange R/O (0 .. 0)
internal.h5	HEX	R/O	HexRange R/O (0 .. 0)
internal.h6	HEX	R/O	HexRange R/O (0 .. 0)
internal.h7	HEX	R/O	HexRange R/O (0 .. 0)
internal.h8	HEX	R/O	HexRange R/O (0 .. 0)
internal.range	TEXT	R/O	StringRange R/O
internal.channel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.target	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tList	TEXT	R/O	StringRange R/O
internal.tSlot	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tName	TEXT	R/O	StringRange R/O
internal.tInit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.r	TEXT	R/O	StringRange R/O
internal.mpapos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mpa0	TEXT	R/O	StringRange R/O
internal.mpa1	TEXT	R/O	StringRange R/O
internal.mpa2	TEXT	R/O	StringRange R/O
internal.mpa3	TEXT	R/O	StringRange R/O
internal.mpa4	TEXT	R/O	StringRange R/O
internal.mpa1pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.mpa2pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.mpa3pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.mpa4pos	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
mpa	FLOAT		DoubleRange (0.000 .. 0.000)
mpa.name	TEXT	R/O	StringRange R/O
mpa.mode	CHOICE		EnumRange (MANUAL AUTOMATIC)
mpa.target	CHOICE		EnumRange (0 1 2 3 4)
mode.az	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
mode.el	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
mode.pl	CHOICE		EnumRange (UNUSED STAND-BY POINTING STOW UNSTOW MANUAL)
mode.mpa	CHOICE		EnumRange (UNUSED STAND-BY POINTING STOW UNSTOW MANUAL)

name	*type*	*flags*	*range*
trackMode	CHOICE		EnumRange (INVALID STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL STOW UNSTOW MANUAL)
subMode	TEXT	R/O	StringRange R/O
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
target	INTEGER		IntegerRange (0 .. 100)
target.name	CHOICE		EnumRange (NONE)
target.frequency	FLOAT	R/O	DoubleRange R/O DIS (0.000000 .. 0.000000)
target.nomLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
target.available	BOOLEAN	R/O	BooleanRange R/O (true false)
target.type	CHOICE	R/O	EnumRange R/O (UNDEFINED TLE I11 TABULAR STAR SUN OTHER)
target.accessible	BOOLEAN	R/O	BooleanRange R/O (true false)
onTarget	BOOLEAN	R/O	BooleanRange R/O (true false)
target.valid	BOOLEAN	R/O	BooleanRange R/O (true false)
target.timeElapsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tle.name	TEXT		StringRange
tle.line1	TEXT		StringRange
tle.line2	TEXT		StringRange
tle.save	TEXT	NOPRESET	StringRange
tle.load	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mpa	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.plVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.start.time	TEXT	R/O	StringRange R/O
pass.end.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.end.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.end.time	TEXT	R/O	StringRange R/O
pass.maxEl.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.maxEl.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pass.maxEl.time	TEXT	R/O	StringRange R/O
pass.valid	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.beacon1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.beacon2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.delta1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.delta2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dtrLock	CHOICE	R/O	EnumRange R/O (UNLOCKED LOCKED)
channel	CHOICE		EnumRange DIS (INVALID A B)
channel.mode	CHOICE		EnumRange DIS (MANUAL AUTOMATIC)
frequency	FLOAT	R/O	DoubleRange R/O DIS (0.000000 .. 0.000000)
freqBand	CHOICE		EnumRange DIS (S-BAND X-BAND)
dtr.control	CHOICE		EnumRange DIS (UNDEFINED NOT IN CONTROL IN CONTROL)
doppler.timestamp	TEXT	R/O	StringRange R/O DIS
doppler.freqOffset	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
doppler.velocity	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
stop	TEXT	NOPRESET	StringRange
jog	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
pointing	TEXT	NOPRESET	StringRange
monopulse	TEXT	NOPRESET	StringRange
monopulseCal	TEXT	NOPRESET	StringRange
steptrack	TEXT	NOPRESET	StringRange
adaptivetrack	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
programtrack	TEXT	NOPRESET	StringRange
scan	TEXT	NOPRESET	StringRange
autotrack	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange
slew	TEXT	NOPRESET	StringRange
mp.modelType	CHOICE		EnumRange (WHITE-NOISE SINGER)
mp.processNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.maneuverTime	FLOAT		DoubleRange (0.000 .. 0.000)
mp.maxAzimuthSpeed	FLOAT		DoubleRange (0.000 .. 0.000)
mp.measurementNoiseSD	FLOAT		DoubleRange (0.000000 .. 0.000000)
mp.signalThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.errorThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.signalFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.distanceFailureThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.modelResetThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.status	TEXT	R/O	StringRange R/O
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.dtrLock	TEXT	R/O	StringRange R/O
mpc.patternType	CHOICE		EnumRange DIS (CIRCULAR)
mpc.azReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)

name	*type*	*flags*	*range*
mpc.elReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK ADAPTIVE LAST-SELECTED)
mpc.direction	CHOICE		EnumRange DIS (CCW CW)
mpc.patternStart	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.patternRadius	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfMeasurements	INTEGER		IntegerRange DIS (0 .. 0)
mpc.signalThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.startThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfSamples	INTEGER		IntegerRange DIS (0 .. 0)
mpc.stabilizeDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.measurementDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.xelevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.xelevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.status	TEXT	R/O	StringRange R/O DIS
mpc.xelevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
mpc.phaseError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.directionError	TEXT	R/O	StringRange R/O DIS
mpc.residual	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.phasingMode	CHOICE		EnumRange DIS (MANUAL AUTOMATIC)
mpc.minFreqSpacing	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
mpc.entryWarningTime	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.entryAlarmTime	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.delete	INTEGER		IntegerRange (0 .. 1)
mpc.getslots	INTEGER		IntegerRange (0 .. 1)
mpc.index.l	INTEGER		IntegerRange (-1 .. 100)
mpc.index.r	INTEGER		IntegerRange (-1 .. 100)
mpc.curpoll	INTEGER		IntegerRange (-1 .. 2)
mpc.curpolr	INTEGER		IntegerRange (-1 .. 2)
mpc.activepol	INTEGER		IntegerRange (-1 .. 2)
st.algorithm	CHOICE		EnumRange DIS (BASIC GRADIENT)
st.maxIterations	INTEGER		IntegerRange DIS (0 .. 0)
st.corrThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.cycleTime	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.beamWidth	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.startThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
st.signalThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.optThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.sequence	CHOICE		EnumRange DIS (CROSS-ELEV ELEVATION)
st.direction	CHOICE		EnumRange DIS (DEC1-DEC2 DEC1-INC2 INC1-INC2 INC1-DEC2)
st.noOfSamples	INTEGER		IntegerRange DIS (0 .. 0)
st.stepSize	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.stabilizeDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.measurementDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.model	CHOICE		EnumRange DIS (NO-MODEL MODEL-1 MODEL-2 MODEL-3)
st.decayRate0	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.decayRate1	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.decayRate2	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.decayRate3	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.status	TEXT	R/O	StringRange R/O DIS
st.xelevationOffs	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.elevationOffs	FLOAT		DoubleRange DIS (0.000 .. 0.000)
st.currentModel	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
st.currentDistance	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.samplingPeriod	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.positionNoiseSD	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.speedNoiseSD	FLOAT		DoubleRange DIS (0.00000 .. 0.00000)
at.deviationNoiseSD	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.okPeriodThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.warningPeriodThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.errorPeriodThreshold	FLOAT		DoubleRange DIS (0.000 .. 0.000)
at.enabled	CHOICE		EnumRange DIS (DISABLED ENABLED)
at.status	TEXT	R/O	StringRange R/O DIS
position.01	CHOICE		EnumRange (A B)
position.02	CHOICE		EnumRange (A B)
position.03	CHOICE		EnumRange (A B)
position.04	CHOICE		EnumRange (A B)
position.05	CHOICE		EnumRange (A B)
position.06	CHOICE		EnumRange (A B)
position.07	CHOICE		EnumRange (A B)
position.08	CHOICE		EnumRange (A B)
position.09	CHOICE		EnumRange (A B)
position.10	CHOICE		EnumRange (A B)
position.11	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
position.12	CHOICE		EnumRange (A B)
position.13	CHOICE		EnumRange (A B)
position.14	CHOICE		EnumRange (A B)
position.15	CHOICE		EnumRange (A B)
position.16	CHOICE		EnumRange (A B)
wgstat.1	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.2	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.3	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.4	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.5	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.6	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.7	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.8	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.9	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.10	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.11	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.12	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.13	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.14	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.15	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.16	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.1	HEX		HexRange (0 .. 0)
wgcmd.2	HEX		HexRange (0 .. 0)
wgcmd.3	HEX		HexRange (0 .. 0)
wgcmd.4	HEX		HexRange (0 .. 0)
wgcmd.5	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
wgcmd.6	HEX		HexRange (0 .. 0)
wgcmd.7	HEX		HexRange (0 .. 0)
wgcmd.8	HEX		HexRange (0 .. 0)
wgcmd.9	HEX		HexRange (0 .. 0)
wgcmd.10	HEX		HexRange (0 .. 0)
wgcmd.11	HEX		HexRange (0 .. 0)
wgcmd.12	HEX		HexRange (0 .. 0)
wgcmd.13	HEX		HexRange (0 .. 0)
wgcmd.14	HEX		HexRange (0 .. 0)
wgcmd.15	HEX		HexRange (0 .. 0)
wgcmd.16	HEX		HexRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power distribution)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Motor 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Motor 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Az/El Encoder)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access door)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Limit switches 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Limit switches 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Hardware)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (General)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF switches)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ACU summary)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Axis summary 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Axis summary 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (SCU summary)
faults.21	ALARM	R/O	AlarmFlagRange R/O (RF Switch 01)
faults.22	ALARM	R/O	AlarmFlagRange R/O (RF Switch 02)
faults.23	ALARM	R/O	AlarmFlagRange R/O (RF Switch 03)
faults.24	ALARM	R/O	AlarmFlagRange R/O (RF Switch 04)
faults.25	ALARM	R/O	AlarmFlagRange R/O (RF Switch 05)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF Switch 06)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RF Switch 07)
faults.28	ALARM	R/O	AlarmFlagRange R/O (RF Switch 08)
faults.29	ALARM	R/O	AlarmFlagRange R/O (RF Switch 09)
faults.30	ALARM	R/O	AlarmFlagRange R/O (RF Switch 10)

name	*type*	*flags*	*range*
faults.31	ALARM	R/O	AlarmFlagRange R/O (RF Switch 11)
faults.32	ALARM	R/O	AlarmFlagRange R/O (RF Switch 12)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RF Switch 13)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RF Switch 14)
faults.35	ALARM	R/O	AlarmFlagRange R/O (RF Switch 15)
faults.36	ALARM	R/O	AlarmFlagRange R/O (RF Switch 16)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Dehydrator)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Tracking LNA)
faults.39	ALARM	R/O	AlarmFlagRange R/O (DNC controller)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Low Elevation)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Beacon Lost)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Beacon Saturated)
faults.43	ALARM	R/O	AlarmFlagRange R/O (DTR in Local)
faults.44	ALARM	R/O	AlarmFlagRange R/O (DTR Summary)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Monopulse Cal Needed)
faults.46	ALARM	R/O	AlarmFlagRange R/O (DTR Configuration)

name	*type*	*flags*	*range*
faults.47	ALARM	R/O	AlarmFlagRange R/O (Time Reference)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Soft Limits Enabled)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Logging Enabled)
faults.50	ALARM	R/O	AlarmFlagRange R/O (IERS EOP EOT warn)
faults.51	ALARM	R/O	AlarmFlagRange R/O (IERS EOP EOT alarm)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 1)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 2)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 3)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Soft End limit axis 4)
faults.56	ALARM	R/O	AlarmFlagRange R/O (DTR Comms Alarm 1)
faults.57	ALARM	R/O	AlarmFlagRange R/O (DTR Comms Alarm 2)
faults.58	ALARM	R/O	AlarmFlagRange R/O (End of Pred Model Valid)
faults.59	ALARM	R/O	AlarmFlagRange R/O (End of Orbit Model Valid)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Below Horizon Mask)
faults.61	ALARM	R/O	AlarmFlagRange R/O (HW Diagnostic Summary)

9.299 Hitec-HACU-01

This page describes the device driver and the device window for the Hitec HACH-01 antenna

controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- Monopulse tracking parameters (only present if configured as monopulse ACU)
- --- Monopulse calibration parameters (only present if configured as monopulse ACU)
- --- Steptrack parameters
- --- Adaptive tracking parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
smallJogDelta	The (small) amount to move the antenna in jog mode.
largeJogDelta	The (large) amount to move the antenna in jog mode.
beaconInTargets	Selects wether to include the beacon frequency in target memories.
operatorPasswd	The operator password, required to communicate with the ACU.
model	Selects if the ACU is a step track or a monopulse model.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.smallJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.largeJogDelta	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 9.90)
config.beaconInTargets	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.model	CHOICE	SAVE SETUP	EnumRange (STEPTRACK MONOPULSE)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.h	HEX	R/O	HexRange R/O (0 .. 0)
internal.v1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.v3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
prev.mode.az	TEXT	R/O	StringRange R/O
prev.mode.el	TEXT	R/O	StringRange R/O
prev.mode.pl	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.login	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
mode.az	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL)
mode.el	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL)
mode.pl	CHOICE		EnumRange (UNUSED STAND-BY POINTING SLEW PROGRAM-TRACK SCAN STEP-TRACK MONOPULSE ADAPTIVE-TRACK AUTOTRACK MONOPULSE-CAL)
state.mode1	TEXT	R/O	StringRange R/O
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.delta1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.delta2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dtrLock	BOOLEAN	R/O	BooleanRange R/O (true false)
stop	TEXT	NOPRESET	StringRange
jog	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
pointing	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
monopulse	TEXT	NOPRESET	StringRange
monopulseCal	TEXT	NOPRESET	StringRange
steptrack	TEXT	NOPRESET	StringRange
adaptivetrack	TEXT	NOPRESET	StringRange
mp.modelType	CHOICE		EnumRange (WHITE-NOISE SINGER)
mp.processNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
mp.maneuverTime	FLOAT		DoubleRange (0.000 .. 0.000)
mp.maxAzimuthSpeed	FLOAT		DoubleRange (0.000 .. 0.000)
mp.measurementNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
mp.signalThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
mp.errorThreshold	FLOAT		DoubleRange (0.000 .. 0.000)
mp.status	TEXT	R/O	StringRange R/O
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mpc.patternType	CHOICE		EnumRange DIS (CIRCULAR)
mpc.azReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK LAST-SELECTED)
mpc.elReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK LAST-SELECTED)
mpc.plReferenceMode	CHOICE		EnumRange DIS (STAND-BY POINTING SLEW PROGRAM-TRACK LAST-SELECTED)
mpc.direction	CHOICE		EnumRange DIS (CCW CW)
mpc.patternStart	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.patternRadius	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.noOfMeasurements	INTEGER		IntegerRange DIS (0 .. 0)

name	*type*	*flags*	*range*
mpc.signalThreshold	FLOAT		DoubleRange DIS (0.0 .. 0.0)
mpc.noOfSamples	INTEGER		IntegerRange DIS (0 .. 0)
mpc.stabilizeDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.measurementDelay	FLOAT		DoubleRange DIS (0.000 .. 0.000)
mpc.xelevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationPos	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.xelevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.status	TEXT	R/O	StringRange R/O DIS
mpc.xelevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.elevationGain	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
mpc.phaseError	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
st.algorithm	CHOICE		EnumRange (BASIC GRADIENT)
st.maxIterations	INTEGER		IntegerRange (0 .. 0)
st.corrThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
st.cycleTime	FLOAT		DoubleRange (0.000 .. 0.000)
st.beamWidth	FLOAT		DoubleRange (0.000 .. 0.000)
st.signalThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
st.optThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
st.sequence	CHOICE		EnumRange (CROSS-ELEV ELEVATION)
st.direction	CHOICE		EnumRange (DEC1-DEC2 DEC1-INC2 INC1-INC2 INC1-DEC2)

name	*type*	*flags*	*range*
st.noOfSamples	INTEGER		IntegerRange (0 .. 0)
st.stepSize	FLOAT		DoubleRange (0.000 .. 0.000)
st.stabilizeDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.measurementDelay	FLOAT		DoubleRange (0.000 .. 0.000)
st.model	CHOICE		EnumRange (NO-MODEL MODEL-1 MODEL-2 MODEL-3)
st.decayRate0	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate1	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate2	FLOAT		DoubleRange (0.000 .. 0.000)
st.decayRate3	FLOAT		DoubleRange (0.000 .. 0.000)
st.status	TEXT	R/O	StringRange R/O
st.xelevationOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
st.elevationOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
st.currentModel	TEXT	R/O	StringRange R/O
st.currentDistance	FLOAT		DoubleRange (0.000 .. 0.000)
at.samplingPeriod	FLOAT		DoubleRange (0.000 .. 0.000)
at.positionNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.speedNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.deviationNoiseSD	FLOAT		DoubleRange (0.000 .. 0.000)
at.status	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power distribution)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Motor 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Motor 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Motor 3)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access door)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Limit switches)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.10	ALARM	R/O	AlarmFlagRange R/O (General)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RF switches)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (ACU summary)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Axis)

9.300 Hitec-HAFP-00-NAOS1

This page describes the monitoring only device driver and the device window for the Hitec antenna front panel including deicing parameters, temperatures and different summary alarms.
This driver version is customized for NAOS project. It is based on HAFP-00_EL_NAOS_AfpInterfaceControlDocument_V1.01 from 20220531. ACU parameters are not included as they are implemented directly in the Hitec-HACU-01-NAOS

The driver uses the Hitec protocol.
IP address and port should be given by the interface in the device setup.
Default communication port is 3000

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The temperature, humidity and weather stattion sensors
- --- The deicing and rainblower control and status parameters
- --- The climatic control and status parameters
- --- The faults page.
- --- The device info page.

- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	No address required
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
temp.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.driveCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.powerCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
wind.speed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
resetESLoop	TEXT	NOPRESET	StringRange
bright.direction	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
bright.lux	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
pressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rain.intensity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
wind.direction	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
deice.status	CHOICE		EnumRange (AUTO MANUAL)
deice.auto	CHOICE		EnumRange (OFF ON)
deice.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.manualAC	TEXT	NOPRESET	StringRange
deice.manualBD	TEXT	NOPRESET	StringRange
deice.manualSub	TEXT	NOPRESET	StringRange
deice.manualCycle	TEXT	NOPRESET	StringRange
deice.heating	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.ac	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.bd	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.subref	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.cycle	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
feedblower.status	CHOICE	R/O	EnumRange R/O (OFF ON)
feedheater.status	CHOICE	R/O	EnumRange R/O (OFF ON)
feedblower	CHOICE		EnumRange (OFF AUTO FORCED ON)
feedheater	CHOICE		EnumRange (OFF AUTO)
feedheater2	CHOICE	R/O	EnumRange R/O (OFF AUTO MANUAL ON)
state.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
climatic.status	CHOICE		EnumRange (ON OFF)
climatic.fan	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
climatic.heater	CHOICE	R/O	EnumRange R/O (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Drive summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (General summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Climatic summary)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Deicing summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Final limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Feed blower fuse)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Feed blower)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Deicing main fuse)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Deicing surge protect)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Deicing auto)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Feed heater)
faults.15	ALARM	R/O	AlarmFlagRange R/O (RF inhibit (fast))
faults.24	ALARM	R/O	AlarmFlagRange R/O (Dehyd low pressure)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Hub Temp/Humidity)
faults.31	ALARM	R/O	AlarmFlagRange R/O (High power load 1)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (High power load 2)
faults.33	ALARM	R/O	AlarmFlagRange R/O (High power load 3)
internal.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.test	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.301 Hitec-HAFP-00

This page describes the monitoring only device driver and the device window for the Hitec antenna front panel including deicing parameters, temperatures and different summary alarms. ACU parameters are not included as they are implemented directly in Hitec-HACU-01 device

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The antenna front panel main page
- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
deice.status	CHOICE		EnumRange (OFF AUTOMATIC)
deice.auto	CHOICE		EnumRange (OFF ON)
deice.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
feedblower.status	CHOICE	R/O	EnumRange R/O (OFF ON)
state.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.a	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.b	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.c	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.d	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.subref	BOOLEAN	R/O	BooleanRange R/O (true false)
temp.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.aped	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.driveCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.powerCabinet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.outsite	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.aped	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
humidity.ahub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Drive Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (General Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Climatic Summary)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Deicing Summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Access Summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Final Limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Rain blower fuse)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Rain blower)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Deicing fuse)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Deicing surge protect)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Deicing off)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.22	ALARM	R/O	AlarmFlagRange R/O (APED 1 Fuse)
faults.23	ALARM	R/O	AlarmFlagRange R/O (APED 2 Fuse)
internal.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.302 Hitec-HSCU-00

This page describes the device driver and the device window for the HITEC Servo Control Unit. ACU parameters are not included as they are implemented directly in Hitec-HACU-01 device

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- Status and sensor readings
- --- Deicing control
- --- Climatic control
- --- Detailed fault flags
- --- Faults and fault mask
- --- Device Info
- --- Maintenance

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
ntpAddress	The IP address of the NTP server the device uses for time synchonisation
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.hasShelterClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPedestalClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasYokeClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasHubClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasFeedClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasDeicing	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasCabinetsClimaticControl	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasOutsideWeatherMonitoring	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasRainBlowing	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasDehydrating	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasRFSafetyLoop	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasMotionWarningLight	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasMotionWarningHorn	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasAircraftWarningLight	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPedestalLight	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasYokeLight	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
info.hasHubLight	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasShelterFireDetection	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPedestalFireDetection	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasYokeFireDetection	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasHubFireDetection	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPedestalDoorOpened	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPedestalDoorClosed	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasYokeDoorOpened	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasYokeDoorClosed	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasHubDoorOpened	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasHubDoorClosed	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasAccessDoorOpened	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasAccessDoorClosed	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasPersonalLift	BOOLEAN	R/O	BooleanRange R/O (true false)
info.hasHandCrank	BOOLEAN	R/O	BooleanRange R/O (true false)
config.ntpAddress	TEXT	SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
rainBlower.mode	CHOICE		EnumRange (OFF AUTOMATIC MANUAL)
deice.temp.min	FLOAT		DoubleRange (0.0 .. 0.0)
deice.temp.max	FLOAT		DoubleRange (0.0 .. 0.0)
deice.humidity.min	FLOAT		DoubleRange (0 .. 0)
deice.humidity.hyst	FLOAT		DoubleRange (0 .. 0)
deice.heat.restart	CHOICE		EnumRange (CONTINUE RESTART)
deice.sect.restart	CHOICE		EnumRange (CONTINUE RESTART)
deice.heat.time	CHOICE		EnumRange (FIXED 30sec VARIABLE)
deice.heat.maxTime	INTEGER		IntegerRange (0 .. 0)
deice.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL MANUAL DEICE)
pedestal.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)
feed.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)

name	*type*	*flags*	*range*
yoke.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)
hub.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)
shelter.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)
cabinet.ccm	CHOICE		EnumRange (DEACTIVATED AUTOMATIC MANUAL)
deice.ref.tset	FLOAT		DoubleRange (0.0 .. 0.0)
deice.sub.tset	FLOAT		DoubleRange (0.0 .. 0.0)
pedestal.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)
pedestal.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
feed.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)
feed.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
hub.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)
hub.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
yoke.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
yoke.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
shelter.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)
shelter.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
cabinet.heat.tset	FLOAT		DoubleRange (0.0 .. 0.0)
cabinet.cool.tset	FLOAT		DoubleRange (0.0 .. 0.0)
deice.ref.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
deice.sub.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
pedestal.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
pedestal.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
feed.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
feed.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
hub.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
hub.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
yoke.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
yoke.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
shelter.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
shelter.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
cabinet.heat.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
cabinet.cool.hyst	FLOAT		DoubleRange (0.0 .. 0.0)
save	TEXT	NOPRESET	StringRange
initialized	TEXT	R/O	StringRange R/O
general.autosave	BOOLEAN	R/O	BooleanRange R/O (true false)
general.standalone	BOOLEAN	R/O	BooleanRange R/O (true false)
shelter.heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
shelter.heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)
shelter.cooling	BOOLEAN	R/O	BooleanRange R/O (true false)
shelter.ventilating	BOOLEAN	R/O	BooleanRange R/O (true false)
pedestal.heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
pedestal.heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)
pedestal.cooling	BOOLEAN	R/O	BooleanRange R/O (true false)
pedestal.ventilating	BOOLEAN	R/O	BooleanRange R/O (true false)
yoke.heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
yoke.heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
yoke.cooling	BOOLEAN	R/O	BooleanRange R/O (true false)
yoke.ventilating	BOOLEAN	R/O	BooleanRange R/O (true false)
hub.heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
hub.heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)
hub.cooling	BOOLEAN	R/O	BooleanRange R/O (true false)
hub.ventilating	BOOLEAN	R/O	BooleanRange R/O (true false)
feed.heating.1	BOOLEAN	R/O	BooleanRange R/O (true false)
feed.heating.2	BOOLEAN	R/O	BooleanRange R/O (true false)
feed.cooling	BOOLEAN	R/O	BooleanRange R/O (true false)
feed.ventilating	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.a	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.b	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.c	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.d	BOOLEAN	R/O	BooleanRange R/O (true false)
deicing.subref	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
rain.detected	BOOLEAN	R/O	BooleanRange R/O (true false)
ice.detected	BOOLEAN	R/O	BooleanRange R/O (true false)
rain.blowing	BOOLEAN	R/O	BooleanRange R/O (true false)
dehydrating	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.lowelovr	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.sspbmute	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.muteswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.sspb1on	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.sspb2on	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.horizonmute	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.autoswmute	BOOLEAN	R/O	BooleanRange R/O (true false)
safety.manualswmute	BOOLEAN	R/O	BooleanRange R/O (true false)
door.pedestal	BOOLEAN	R/O	BooleanRange R/O (true false)
door.yoke	BOOLEAN	R/O	BooleanRange R/O (true false)
door.hub	BOOLEAN	R/O	BooleanRange R/O (true false)
door.access	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
light.pedestal	BOOLEAN	R/O	BooleanRange R/O (true false)
light.yoke	BOOLEAN	R/O	BooleanRange R/O (true false)
light.hub	BOOLEAN	R/O	BooleanRange R/O (true false)
warn.motionlight	BOOLEAN	R/O	BooleanRange R/O (true false)
warn.motionhorn	BOOLEAN	R/O	BooleanRange R/O (true false)
warn.aircraft	BOOLEAN	R/O	BooleanRange R/O (true false)
pmcuconnected	BOOLEAN	R/O	BooleanRange R/O (true false)
state.OutsideTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideWindSpeed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideWindDir	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideBrightness	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideBrightnessDir	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsideAirPress	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.OutsidePrecipIntensity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ShelterTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.ShelterHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.PedestalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.PedestalHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.YokeTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.YokeHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.HubTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.HubHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.HubChillerProbe	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.FeedTemp1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.FeedTemp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSectionATemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSectionBTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSectionCTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSectionDTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSubreflectorTemp1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DeicingSubreflectorTemp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.PowerCabinetTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.DriveCabinetTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ReflectorCabinetTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AFP general)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Climatic ctl summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Misc summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Emergency summary)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Lights summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fire summary)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power summary)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Connections summary)
faultFlags.shelter	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faultFlags.pedestal	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.yoke	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.hub	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.feed	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.deice1	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.deice2	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.cabs	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.weather	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.rain	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.dehyd	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.rfsafety	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.misc1	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.misc2	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.misc3	HEX	R/O	HexRange R/O (0 .. 0)
faultFlags.misc4	HEX	R/O	HexRange R/O (0 .. 0)
i1	TEXT	R/O	StringRange R/O
i2	TEXT	R/O	StringRange R/O
i3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
i4	TEXT	R/O	StringRange R/O
h1	HEX	R/O	HexRange R/O (0 .. 0)
h2	HEX	R/O	HexRange R/O (0 .. 0)
h3	HEX	R/O	HexRange R/O (0 .. 0)
b1	BOOLEAN	R/O	BooleanRange R/O (true false)
b2	BOOLEAN	R/O	BooleanRange R/O (true false)
b3	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideHumiditySensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideWindSpeedSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideWindDirectionSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideBrightnessSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideBrightnessDirectionSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsideAirPressureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasOutsidePrecipitationIntensitySensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasShelterTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasShelterHumiditySensor	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hasPedestalTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasPedestalHumiditySensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasYokeTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasYokeHumiditySensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasHubTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasHubHumiditySensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasHubChillerRegulationTemperatureProbe	BOOLEAN	R/O	BooleanRange R/O (true false)
hasFeedTemperatureSensor1	BOOLEAN	R/O	BooleanRange R/O (true false)
hasFeedTemperatureSensor2	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSectionATemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSectionBTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSectionCTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSectionDTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSubreflectorTemperatureSensor1	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingSubreflectorTemperatureSensor2	BOOLEAN	R/O	BooleanRange R/O (true false)
hasPowerCabinetTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hasDriveCabinetTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasReflectorCabinetTemperatureSensor	BOOLEAN	R/O	BooleanRange R/O (true false)
hasShelterCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasPedestalCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasYokeCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasHubCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasFeedCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasDeicingCcm	BOOLEAN	R/O	BooleanRange R/O (true false)
hasCabinetCcm	BOOLEAN	R/O	BooleanRange R/O (true false)

9.303 IDirect-5000

This page describes the device driver and the device window for the iDirect 5000 satellite modem / IP router. The driver controls the modem parameters of the device and monitors beside the receive quality information the IP routing table.

The driver uses a telnet / debugging console interface of the iDirect 5000 to monitor and control the device. This causes a number of limitations which are itemized at the bottom of this page.

The driver uses a protocol called *IDirect-Telnet* , which works similar to a serial, line oriented protocol running through a terminal server. Hence, the IP address used to control the modem must be set at the device setup dialog instead of a serial interface name.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General parameters

- --- Meter readings / statistics
- --- IP routing table
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the iDirect 5000:

parameter	***description***
user	The user name, the driver shall use for login at the modem. The user name usually is 'admin'.
password	The password for logging in at the modem. The password usually is 'iDirect'.
lnbSupply	Switches the DC power supply for the LNB on/off.
lnb10MHz	Switches the 10 MHz reference for the LNB on/off.
lnbLOFreqLo	Defines the lower band LO frequency
lnbLOFreqHi	Defines the higher band LO frequency
lnbLOEdge	Defines the edge between low and high band
bucSupply	Switches the DC power supply for the BUC on/off.
buc10MHz	Switches the 10 MHz reference for the BUC on/off.
bucLOFreq	Defines the BUC LO frequency

Remarks

- A couple of parameters only can be commanded at the modem by modifying it's boot-up configuration file and then rebooting the modem. These parameters are:
  - tx.mod.type (BPSK, QPSK, 8PSK)
  - tx.mod.fec (0.431, 0.495, 0.533, 0.793, 0.879)
  - tx.mod.spectrum (NORM, INVERT)
  - rx.dem.type (BPSK, QPSK, 8PSK)
  - rx.dem.fec (0.431, 0.495, 0.533, 0.793, 0.879)
  - rx.dem.spectrum (NORM, INVERT) Changing one of these parameters means to reboot the modem which in fact interrupts the traffic (and the control communication) for about 30 seconds.
- The modem does not preserve 'tx.on' and 'rx.on' when switched off or rebooted. These parameters are stored at the VLC instead and set to the stored value as soon as the the communication to the modem gets established.In fact, the modem always will start

transmitting for a short time when powered on until the VLC is capable to switch this off (if OFF is the stored setting). While during a reboot this will happen only for a couple of milliseconds, the modem may stay transmitting for several seconds if both, the modem and the VLC are powered on at the same time. The VLC will take a longer time to boot, the modem will already be transmitting when the VLC application starts.

- To see the rx/tx frequency values in terms of L-band frequency, enter 0.0 for all lnbLOFreq / bucLOFreq configuration settings.
- To configure the modem driver for a single band LNB, set both lnbLOFreq configuration parameters to the LNB's LO frequency, the lnbLOEdge parameter to zero.
- On power on, the driver always assumes that a frequency value in the low (e.g. 11GHz) band is set. Hence, if a receive frequency in the upper band was selected before the VLC was powered down, it will come up with a wrong receive frequency. This does not disturb the signal reception, the VLC simply applies the wrong LO frequency to display the value. Once a receive frequency has been commanded by the operator, the VLC remembers the correct frequency band until power down.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.application	TEXT	R/O	StringRange R/O
info.buildLabel	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.did	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.rebooting	BOOLEAN	R/O	BooleanRange R/O (true false)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
config.user	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
config.lnbSupply	CHOICE	SETUP	EnumRange (OFF ON)
config.lnb10MHz	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbLOFreqLo	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)
config.lnbLOFreqHi	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)
config.lnbLOEdge	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)
config.bucSupply	CHOICE	SETUP	EnumRange (OFF ON)
config.buc10MHz	CHOICE	SETUP	EnumRange (OFF ON)
config.bucLOFreq	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.j	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.k	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a	TEXT	R/O	StringRange R/O
internal.ftx	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.frx	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.rxband	TEXT	R/O	StringRange R/O
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35 .. 7)
tx.on	CHOICE	SAVE	EnumRange (OFF ON)
tx.mod.dataRate	INTEGER		IntegerRange (64000 .. 4200000)
tx.mod.symbolRate	INTEGER	R/O	IntegerRange R/O (4800 .. 2875000)
tx.mod.type	CHOICE		EnumRange (BPSK QPSK 8PSK)

name	*type*	*flags*	*range*
tx.mod.fec	CHOICE		EnumRange (0.431 0.495 0.533 0.793 0.879)
tx.mod.blocksPerFrame	INTEGER		IntegerRange (0 .. 0)
tx.mod.spectrum	CHOICE		EnumRange (NORM INVERT)
tx.mod.cwMode	CHOICE		EnumRange (MOD C/W)
rx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
rx.on	CHOICE	SAVE	EnumRange (OFF ON)
rx.dem.dataRate	INTEGER		IntegerRange (128000 .. 18000000)
rx.dem.symbolRate	INTEGER	R/O	IntegerRange R/O (64000 .. 11500000)
rx.dem.type	CHOICE		EnumRange (BPSK QPSK 8PSK)
rx.dem.fec	CHOICE		EnumRange (0.431 0.495 0.533 0.793 0.879)
rx.dem.blocksPerFrame	INTEGER		IntegerRange (0 .. 0)
rx.dem.spectrum	CHOICE		EnumRange (NORM INVERT)
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.lnbVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.meter.bucVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.meter.bucCurrent	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ip.routingTable	TEXT	R/O	StringRange R/O
internal.save	TEXT	R/O	StringRange R/O
internal.reboot	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RX lock)

9.304 IP-Sens-IPS2225P

This page describes the device driver and the device window for*IP-Sens-IPS2225P* temperature and humidity sensor via SNMP. The driver can handle up to 5 sensors.

It was tested with Model IPS2225 and 4 sensors.

- FW 1.04,
- HW version 3

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sensor readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
sensors	Defines the number pf installed sensor to be monitored.
sensor.1..4.loTemp	Defines the low temperature threshold in °C. If this value is undercut a low temperature alarm is raised
sensor.1..4.hiTemp	Defines the high temperature threshold in °C. If this value is exceeded a high temperature alarm is raised
sensor.1..4.loHumidity	Defines the low humidity threshold in %. If this value is undercut a low humidity alarm is raised
sensor.1..4.hiHumidity	Defines the high humidity threshold in %. If this value is exceeded a high humidity alarm is raised
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.product	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.unit	TEXT	R/O	StringRange R/O
info.sensorId1	TEXT	R/O	StringRange R/O
info.sensorId2	TEXT	R/O	StringRange R/O
info.sensorId3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.sensorId4	TEXT	R/O	StringRange R/O
info.sensorId5	TEXT	R/O	StringRange R/O
config.sensors	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5)
config.sensor.1.loTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.1.hiTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.1.loHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.1.hiHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.2.loTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.2.hiTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.2.loHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.2.hiHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.3.loTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.3.hiTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.3.loHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.3.hiHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.4.loTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.4.hiTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.4.loHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.4.hiHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.5.loTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.5.hiTemp	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.5.loHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.sensor.5.hiHumidity	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

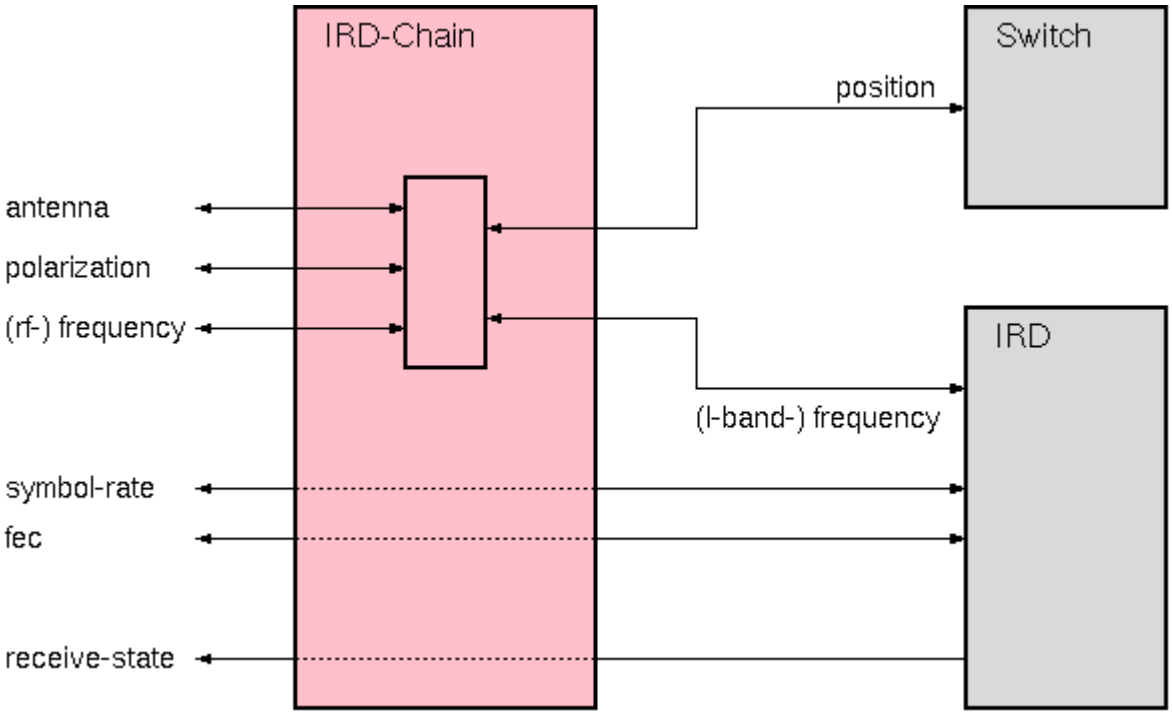
name	*type*	*flags*	*range*
temperature.1f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.4f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.5f	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
sensor.1	TEXT		StringRange
sensor.2	TEXT		StringRange
sensor.3	TEXT		StringRange
sensor.4	TEXT		StringRange
sensor.5	TEXT		StringRange
reset	BOOLEAN	NOPRESET	BooleanRange (true false)
internal.t	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 Low Temp)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 High Temp)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 Low Humidity)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 High Humidity)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 Low Temp)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 High Temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 Low Humidity)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 High Humidity)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 Low Temp)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 High Temp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 Low Humidity)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 High Humidity)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 Low Temp)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 High Temp)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 Low Humidity)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 High Humidity)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 Low Temp)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 High Temp)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 Low Humidity)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 High Humidity)

9.305 IRD-Chain

The IRD-Chain logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization. The IRD-Chain device is similar to the Band-Pol-4-To-1 device, but it is much more flexible to configure. It primarily is intended for configurations where several IRDs are connected to an L-band switch matrix which distributes the signals from multiple antennas. It however can be used for simpler applications, too.



The IRD-Chain device translates the user level parameters 'antenna', 'polarization' and 'frequency' into the frequency setting at the IRD and the position of the input selection switch / matrix output. The translation is made following the rules defined in a configuration file. This file describes the switch matrix inputs. The IRD-Chain device does the translation of LO-frequencies, as these may vary for the different antennas.

Beside the function described above, the IRD-Chain device implements a locking mechanism which prevents users from changing parameters at a receive chain which is currently in use. This protection only works on the parameters controlled by the IRD-Chain device, it does not influence the operation of the IRD through it's own screen.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control page for the device.
- --- The fault page
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
  - **IRD Summary** : The associated IRD shows a summary fault (includes Lock fault)
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the IRD-Chain device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
lockByIP	If you set this parameter to 'YES', only the operator who 'locked' the IRD-Chain may unlock it again. He is required to do this from the same workplace whre he locked the device. When set to 'NO' any operator logged in with privilege level 100 or more may unlock a locked IRD-Chain from any workplace.
configFile	The name of the configuration file, the IRD-Chain reads the matrix input assignments from. You may use a common configuration file for all IRD-Chain devices in the system or an individual file. The file format is described at the end of this chapter.

Configuring an M&C with IRD-Chain devices

Unlike other logical devices, which know all links to other parameters / devices from configuration settings, the IRD-Chain device refers to the IRD it controls by it's name. This mainly is because of the IRD device icon which is contained in the IRD-Chain's device window.

The name of the IRD controlled by an IRD-Chain device always must start with an "I" followed by the suffix contained in the IRD-Chain name. The suffix is everything including and following the last hyphen in the name or any trailing digits. The list below shows a few examples for explanation:

IRD-Chain name	***IRD name***

IRD-Chain name	*IRD name*
CHAIN1	I1
CHN-SCOPUS4	I-SCOPUS4
CH-RCV-T3	I-T3
CHAIN	ICHAIN

When you select a name for the IRD-Chain device, you **must** name the IRD following this scheme to make the IRD being controlled by the IRD-Chain.

Using the lock/unlock mechanism in a task-oriented screen

The IRD-Chain's lock/unlock mechanism works through the variables 'locked' and 'togglelock'.

To make a parameter in a task-oriented screen 'protected' by the IRD-Chain device (i.e. the parameter only can be changed it the IRD-Chain is 'unlocked'), set the 'locked' flag of the IRD-Chain as the 'enable ID' of the parameter. Example: you want to protect the 'program' parameter of IRD I-3. The chain device is named CHN-3. Set the 'enable ID' field for the 'program' parameter to 'CHN-3.locked' and the 'enable Val' to 'false'. The parameter will be grayed out and treated as read-only as soon as the chain device is locked.

To control the locked state of a IRD-Chain device from a task-oriented screen, introduce a Parameter-Button element which sends the special value '%IP%' to the 'togglelock' variable of the IRD-Chain device. The user interface expands the '%IP%' to string 'username@ip-address' which is used to identify username and workplace from where the IRD-Chain device gets locked.

The configuration file format

The IRD-Chain's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all IRD-Chain devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	*contents*	*remark*
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters ('

column	*contents*	*remark*
3	polarization	You may name the polarization planes with X/Y, H/V, HOR/VERT or what ever mnemonics you prefer. The only restriction is that within one M&C system the polarization planes must be names always the same. Beside this you should consider, that longer tokens may not fit into the choice-box provided by the IRD-Chain's device window.
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	**<** or **>**	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the nect column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#
#  Example configuration file for a IRD-Chain device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |               |
#  |   antenna  |  LO freq. [MHz] |
#  |   |       | |             |

  1   ARD     H  9750.0       < 11850.0
  2   ARD     V  9750.0       < 11850.0
  3   ARD     H  10600.0       > 11850.0
  4   ARD     V  10600.0       > 11850.0

  5   NDR     H  9750.0       < 11850.0
  6   NDR     V  9750.0       < 11850.0
  7   NDR     H  10600.0       > 11850.0
  8   NDR     V  10600.0       > 11850.0

  9   EBU     H  10000.0       > 0.0
 10   EBU     V  10000.0       > 0.0

 11   TTC     H  10000.0       > 0.0
 12   TTC     V  10000.0       > 0.0

 13   ASTRA    H  9750.0       < 11850.0
 14   ASTRA    V  9750.0       < 11850.0
 15   ASTRA    H  10600.0       > 11850.0
 16   ASTRA    V  10600.0       > 11850.0

 17   SNGR    H  9750.0       < 11850.0
 18   SNGR    V  9750.0       < 11850.0
 19   SNGR    H  10600.0       > 11850.0
 20   SNGR    V  10600.0       > 11850.0

 21   I605    V  9750.0       > 0.0

 22   W1      V  10000.0       > 0.0
```

The IRD-Chain device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all IRD-CHain devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.configFile	TEXT	SAVE SETUP	StringRange
antenna	CHOICE	NOPRESET	EnumRange (?)
polarization	CHOICE		EnumRange (H V)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
fec	CHOICE		EnumRange (?)
satellite	TEXT	SAVE	StringRange
channel	TEXT	SAVE	StringRange
alias	TEXT	SAVE NOPRESET	StringRange
quality	TEXT	R/O	StringRange R/O
signal	BOOLEAN	R/O	BooleanRange R/O (true false)
locked	TEXT	R/O	StringRange R/O
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IRD summary)

9.306 Infratec-PM

This page describes the device driver and the device window for the Infratec Power Module IP Switch series and was tested with : PM4-IP (4 ports), PM4-IPS (4 ports), PM8-IP (8 ports), PM8-IPS (8 ports), PM-IP GOTO (8 port rackmount), PM821-IP (8 ports).

The driver detects automatically the connected switch model and adjusts the number of available ports/outlets.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows and commands port status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- retries --- The number of communication retries which are made at application level.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
config.pulseTime.01	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.02	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.03	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.04	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.05	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.06	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.07	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.pulseTime.08	INTEGER	SETUP	IntegerRange DIS (1 .. 36000)
config.name.01	TEXT	SETUP	StringRange DIS
config.name.02	TEXT	SETUP	StringRange DIS
config.name.03	TEXT	SETUP	StringRange DIS
config.name.04	TEXT	SETUP	StringRange DIS
config.name.05	TEXT	SETUP	StringRange DIS
config.name.06	TEXT	SETUP	StringRange DIS
config.name.07	TEXT	SETUP	StringRange DIS
config.name.08	TEXT	SETUP	StringRange DIS
name.01	TEXT	R/O	StringRange R/O DIS
name.02	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
name.03	TEXT	R/O	StringRange R/O DIS
name.04	TEXT	R/O	StringRange R/O DIS
name.05	TEXT	R/O	StringRange R/O DIS
name.06	TEXT	R/O	StringRange R/O DIS
name.07	TEXT	R/O	StringRange R/O DIS
name.08	TEXT	R/O	StringRange R/O DIS
state.01	CHOICE		EnumRange DIS (ON OFF)
state.02	CHOICE		EnumRange DIS (ON OFF)
state.03	CHOICE		EnumRange DIS (ON OFF)
state.04	CHOICE		EnumRange DIS (ON OFF)
state.05	CHOICE		EnumRange DIS (ON OFF)
state.06	CHOICE		EnumRange DIS (ON OFF)
state.07	CHOICE		EnumRange DIS (ON OFF)
state.08	CHOICE		EnumRange DIS (ON OFF)
pulse.01	TEXT	NOPRESET	StringRange DIS
pulse.02	TEXT	NOPRESET	StringRange DIS
pulse.03	TEXT	NOPRESET	StringRange DIS
pulse.04	TEXT	NOPRESET	StringRange DIS
pulse.05	TEXT	NOPRESET	StringRange DIS
pulse.06	TEXT	NOPRESET	StringRange DIS
pulse.07	TEXT	NOPRESET	StringRange DIS
pulse.08	TEXT	NOPRESET	StringRange DIS
internal.mib	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Model not supported)

9.307 Inmarsat-NLES-Status

The Inmarsat-NLES-Status device collects a number of status variables from the M&C and builds a "NLES Status" string from these. The status string is sent to a serial interface every 10 minutes or if one of the input parameters changes.

The actual state of all input parameters may ve monitored at the info page of the device screen, the parameter IDs of the values to monitor are defined at the device setup page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page, shows the actual input values
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window the parameter IDs of the status variables to monitor must be set:

parameter	***description***
hpaNoId	ID of the parameter reporting this information
hpaFaultId	ID of the parameter reporting this information
kuTrackFaultId	ID of the parameter reporting this information
l1LnaNoId	ID of the parameter reporting this information
l1LnaFaultId	ID of the parameter reporting this information
l5LnaNoId	ID of the parameter reporting this information
l5LnaFaultId	ID of the parameter reporting this information

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hpaNo	TEXT	R/O	StringRange R/O
info.hpaFault	TEXT	R/O	StringRange R/O
info.kuTrackFault	TEXT	R/O	StringRange R/O
info.l1LnaNo	TEXT	R/O	StringRange R/O
info.l1LnaFault	TEXT	R/O	StringRange R/O
info.l1AutoTrack	TEXT	R/O	StringRange R/O
info.l5LnaNo	TEXT	R/O	StringRange R/O
info.l5LnaFault	TEXT	R/O	StringRange R/O
info.l5AutoTrack	TEXT	R/O	StringRange R/O
info.ulLineAmpNo	TEXT	R/O	StringRange R/O
info.ulLineAmpFault	TEXT	R/O	StringRange R/O
info.l1LineAmpNo	TEXT	R/O	StringRange R/O
info.l1LineAmpFault	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.l5LineAmpNo	TEXT	R/O	StringRange R/O
info.l5LineAmpFault	TEXT	R/O	StringRange R/O
config.hpaNoId	TEXT	SAVE SETUP	StringRange
config.hpaFaultId	TEXT	SAVE SETUP	StringRange
config.kuTrackFaultId	TEXT	SAVE SETUP	StringRange
config.l1LnaNoId	TEXT	SAVE SETUP	StringRange
config.l1LnaFaultId	TEXT	SAVE SETUP	StringRange
config.l5LnaNoId	TEXT	SAVE SETUP	StringRange
config.l5LnaFaultId	TEXT	SAVE SETUP	StringRange
triggerSend	TEXT	R/O	StringRange R/O
value	TEXT	R/O	StringRange R/O

9.308 Input

This page describes the device driver and the device window for the*Input* logical device. This device monitors one single input port of a PLC device and optionally associates a fault with this input port.

The *Input* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

The *Input* device may be configured in various ways to match the requirements of almost all applications. Key features of the device are:

- The reported port states (default '0' / '1') may be translated to arbitrary strings (e.g. 'Door open' / 'Door closed').
- The device optionally may be configured to generate a fault if either the port is high or low.
- The description of the fault flag is customizable.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / the (translated) PLC port state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- inputPort --- The address of the PLC port to monitor. See the paragraph 'Port addressing' below for details.
- positions --- By default the device reports the state of the input as '0'/'1'. You may specify arbitrary texts for these states. 'positions' contains the two descriptions, separated by a comma. If you customize this setting, you should consider that other logical devices (e.g. a protection switch) may expect standard values like 0, 1, OFF or ON when reading this value.
- faultMonitoring --- Defines for which input states the device shall report a fault.
  - NONE --- Dont' report a fault
  - FAULTIF0 --- Report a fault if the input (physically) reads '0'.
  - FAULTIF1 --- Report a fault if the input (physically) reads '1'.
- faultText --- The text of the fault message. In the event log, changes of the fault state appear as ' ... text ... FAULT' or ' ... text ... OK'.
- commFaultId --- By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault. The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).

Fault flags

fault flag	***fault name***	***description***

fault flag	***fault name***	***description***
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Circuit	This fault is set if the input port state matches the condition configured with the 'faultMonitoring' parameter. The name of this fault may be customized by means of the 'faultText'' parameter.
faults.02	Configuration Setup	This fault signals that the 'inputPort' parameter ist not properly configured. The device expects, that 'inputPort' is set to a value which matches the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the**+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address '*FANUC.I00065+1* ' refers to the *2^1* bit in the variable '*I00065* ' provided by the ' *FANUC* ' device. In fact this addresses the input table register '%I00066' at the PLC. Other PLC drivers may use a different naming scheme for the variables, but the concept always is the same.

'initialized' condition

During the initialization phase of the M&C the state of physical inputs may be not yet known. To prevent the Input device from raising a fault during this phase, the Input device expects that the PLC driver provides an 'initialized' variable which is set true after the phsyical inputs have been read the first time after system start. The Input device will not raise a fault until it recognizes the PLC driver to be initialized.

The Input device derives the device name of the PLC from the configured input port address

and reads the '.initialized' variable to get this state.

All PLC and IO-Controller drivers supplied by SatService GmbH provide this 'initialized' status variable. If you are using the Input device with other device types, be sure that the device provides this 'initialized' value or the Input device will not work as expected.

Remarks

- The *Input* device typically is used with PLC input ports. It however is not restricted to this. One application where the monitoring of output ports may make sense is the interrogation of ports which are controlled by program running in the PLC device itself.
- Using the 'positions' configuration setting you may reverse the polarity of the input. Set it to '1,0' and the device reports the state inverted. Please note, that the setting 'faultMonitoring' always refers to the physical state not to the translated one.
- The *Input* device works completely event driven, it introduces no noticeable delay between the change of the input and the state report. For a fast response you should tune the device thread IDLE time of the thread running the PLC driver to the lowest possible value which does not raise the overall CPU load above 20%. (recommended IDLE time for PLC drivers: 200ms)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputPort	TEXT	SAVE SETUP	StringRange
config.positions	TEXT	SAVE SETUP	StringRange
config.faultMonitoring	CHOICE	SAVE SETUP	EnumRange (NONE FAULT IF 0 FAULT IF 1)
config.faultText	TEXT	SAVE SETUP	StringRange
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Circuit)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.309 Irt-Pico-BUC-RCP

This page describes the device driver and the device window for the IRT Pico BUC Series via RCP over SNMP

The driver was tested with the following models:

- 10dBm Ka-Band Model - FW 1.8.2

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- bucNumber --- Defines the identifier of the BUC in a 1:1 redundancy A=1,B=2

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.gainMin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.gainMax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.attnMin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.attnMax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.loFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.freqMin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.freqMax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.txPowerupState	CHOICE	R/O	EnumRange R/O (ALWAYS ON ALWAYS OFF LAST SAVED STATE UNDEFINED)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.bucNumber	CHOICE	SAVE SETUP	EnumRange (1 2)
config.gainThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange (MANUAL AUTO)
redundancyMode	CHOICE		EnumRange (COLD STANDBY HOT STANDBY)
online	CHOICE		EnumRange (ONLINE)
location	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
position	CHOICE	R/O	EnumRange R/O (A B)
onSwitch	CHOICE	R/O	EnumRange R/O (UNKNOWN BUC-A BUC-B)
internal.bucNumber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bucNumber2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bucNumber3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RCP-BUC Communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (PLL lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Under current)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Hardware)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Redundancy)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Redundancy2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low gain)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.310 Janitza-UMG604

This page describes the device driver and the device window for the Janitza UMG604 line voltage monitor.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first page shows the actual meter readings of the device. Displayed are voltage, current, power and the consumed energy for each of the three phases L1/L2/L3. Power and energy are shown as a total value es well. Current, power and energy readings refer to the active current.
- --- The second page lets you define min / max limits for the voltage readings for all three phases. If a voltage exceeds a limit, a fault is raised and a red light beside the voltage reading at the readings page goes on.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The only setup parameter which is required to operate line voltage monitor, is the IP address of the device.

Remarks

- The voltage limits checked by the device driver are checked in the sat-nms software only. These limits do not have any relation to the limit / transient checking in the UMG604 itself.
- There is no way to reset the energy counters in the UMG604 from the sat-nms software.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysDescr	TEXT	R/O	StringRange R/O
uL1N	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
uL2N	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
uL3N	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
iL1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
iL2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
iL3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
pL1	FLOAT	R/O	DoubleRange R/O (0 .. 0)
pL2	FLOAT	R/O	DoubleRange R/O (0 .. 0)
pL3	FLOAT	R/O	DoubleRange R/O (0 .. 0)
pTotal	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.p	FLOAT	R/O	DoubleRange R/O (0 .. 0)
eL1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
eL2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
eL3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
eTotal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
uL1LowLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
uL2LowLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
uL3LowLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
uL1HighLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
uL2HighLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
uL3HighLimit	FLOAT	SAVE	DoubleRange (0.0 .. 400.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (L1 voltage low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (L1 voltage high limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (L2 voltage low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (L2 voltage high limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (L3 voltage low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (L3 voltage high limit)

9.311 KBR-Multimess-4F96

This page describes the device driver and the device window for the*KBR Multimess 4F96* mains power measurement device. The driver reads the following measurement values from the device for all three lines of a three-phase voltage:

- Voltage to neutral
- Apparent power
- Active power
- Reactive power
- Active factor / cos(phi) The driver uses Modbus/TCP to read out the measurement values, hence the device has to be equipped with this communication option.

Configuration settings

The KBR Multimess unit is controlled through Modbus/TCP. The device's IP address must be set

as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual measurement values
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerUnit	CHOICE	SAVE SETUP	EnumRange (W kW MW)
voltage.l1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
voltage.l2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.l3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
apparentPower.l1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
apparentPower.l2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
apparentPower.l3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
activePower.l1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
activePower.l2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
activePower.l3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
reactivePower.l1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
reactivePower.l2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
reactivePower.l3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
cosPhi.l1	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
cosPhi.l2	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
cosPhi.l3	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

9.312 Key-Generator

The logical device Key-Generator generates random keys/passphrases with 16, 32, 48 and 64 hex digits. The generator is using the kernel random number source device from a linux system. The script hex-random in directory /scripts is called by this logical device via the button generate. The keys are used by copy and past to other devices or into other applications e.g. mail program. The generated keys could be used to encrypt a transmission.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the keys and the generate button.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
key.16	TEXT		StringRange
key.32	TEXT		StringRange
key.48	TEXT		StringRange
key.64	TEXT		StringRange
generate	TEXT	NOPRESET	StringRange

9.313 Keysight-FieldFox

This page describes the Keysight-FieldFox (Spectrum-Analyzer) device driver and the device window for this device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like

center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

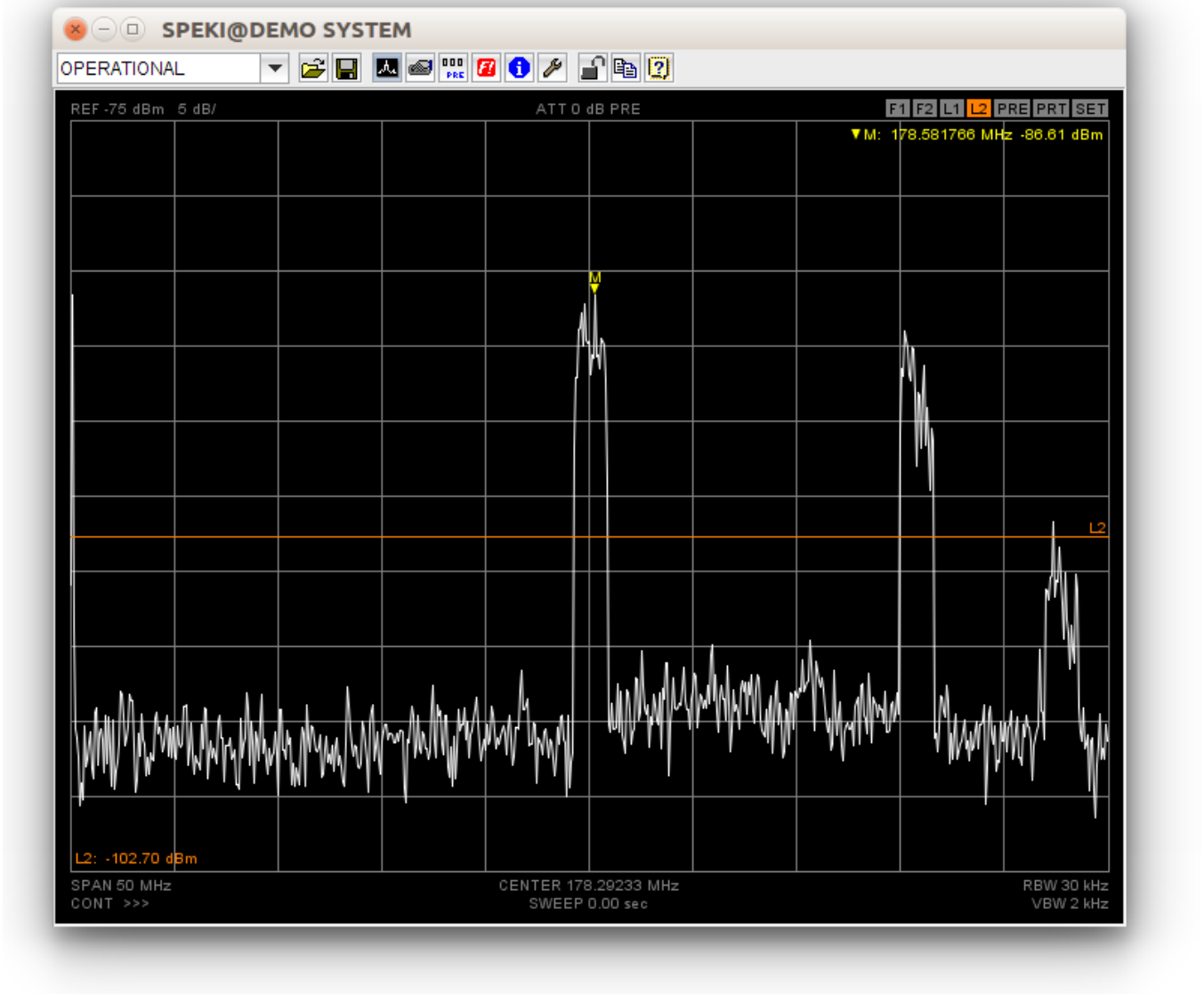
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Keysight-FieldFox driver provides the two presets*Factory-* and *User-Preset*. You can recall both presets but you can only save user preset. The Presets are save directly on the driver.

How to configure the driver

The Keysight-FieldFox are controlled via a TCP connection at port 5024. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5024. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwrevision	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0 .. 0)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 10000.000000)
span	FLOAT		DoubleRange (0.000000 .. 10000.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 10000.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 10000.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT	R/O	DoubleRange R/O (0.004 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE	R/O	EnumRange R/O (AUTO)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	R/O	StringRange R/O
memname.1	TEXT	R/O	StringRange R/O
memname.2	TEXT	R/O	StringRange R/O
memname.3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
memname.4	TEXT	R/O	StringRange R/O
memname.5	TEXT	R/O	StringRange R/O
memname.6	TEXT	R/O	StringRange R/O
memname.7	TEXT	R/O	StringRange R/O
memname.8	TEXT	R/O	StringRange R/O
memname.9	TEXT	R/O	StringRange R/O
memname.10	TEXT	R/O	StringRange R/O
memname.11	TEXT	R/O	StringRange R/O
memname.12	TEXT	R/O	StringRange R/O
memname.13	TEXT	R/O	StringRange R/O
memname.14	TEXT	R/O	StringRange R/O
memname.15	TEXT	R/O	StringRange R/O
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

9.314 LPT-3000RX4

This page describes the LPZ 3000RX4 (Spectrum-Analyzer) device driver and the device window for this device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

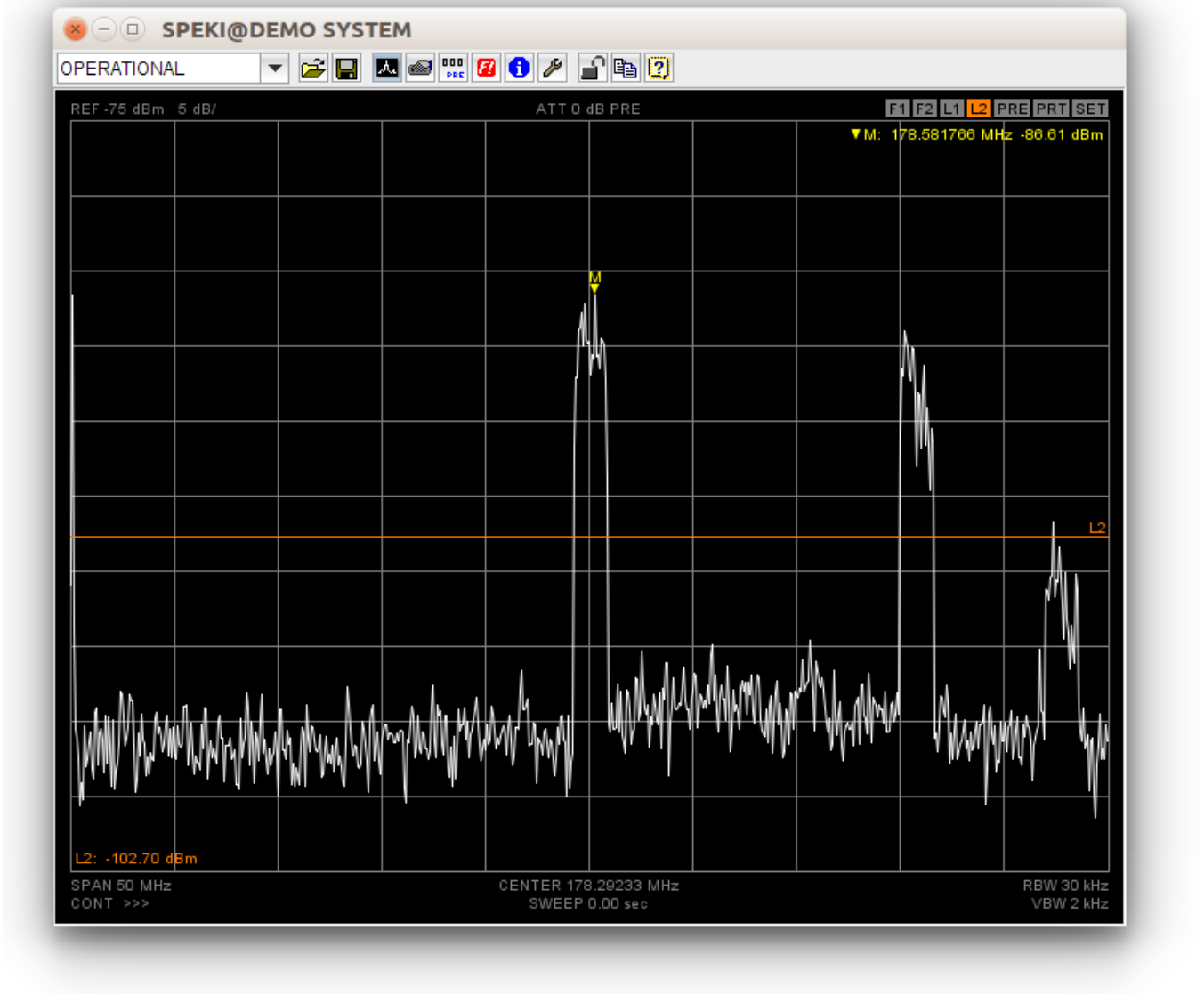
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The LPT-3000RX4 driver provides the two presets *Factory-* and *User-Preset*. You can recall both presets but you can only save user preset. The Presets are save directly on the driver.

How to configure the driver

The LPT-3000RX4 are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwrevision	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	INTEGER		IntegerRange (1 .. 4)
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE	R/O	EnumRange R/O (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	R/O	StringRange R/O
memname.1	TEXT	R/O	StringRange R/O
memname.2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
memname.3	TEXT	R/O	StringRange R/O
memname.4	TEXT	R/O	StringRange R/O
memname.5	TEXT	R/O	StringRange R/O
memname.6	TEXT	R/O	StringRange R/O
memname.7	TEXT	R/O	StringRange R/O
memname.8	TEXT	R/O	StringRange R/O
memname.9	TEXT	R/O	StringRange R/O
memname.10	TEXT	R/O	StringRange R/O
memname.11	TEXT	R/O	StringRange R/O
memname.12	TEXT	R/O	StringRange R/O
memname.13	TEXT	R/O	StringRange R/O
memname.14	TEXT	R/O	StringRange R/O
memname.15	TEXT	R/O	StringRange R/O
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.315 Leitch-Panacea

This page describes the device driver and the device window for a Leitch ASI Switch Matrix (Panacea series) with up to 8 inputs and outputs. The driver supports only interfaces on Level 0.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Matrix do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
info.bootloader	TEXT	R/O	StringRange R/O
info.module	TEXT	R/O	StringRange R/O
info.cntr.serial	TEXT	R/O	StringRange R/O
info.inf.serial	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)

name	*type*	*flags*	*range*
output.02	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.03	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.04	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.05	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.06	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.07	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
output.08	CHOICE		EnumRange (0001 0002 0003 0004 0005 0006 0007 0008)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
reset	TEXT		StringRange
internal.manual	TEXT	R/O	StringRange R/O
internal.board	TEXT	R/O	StringRange R/O
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.316 Level-Add

The Level-Add logical device provides a simple mechanism to monitor transmit power values in EIRP. The Level-Add device uses a fixed calibration offset. It work like the Level-Set device but only for reading the EIRP.

The Level-Add device monitors a configurable parameter for power measurement and adds a calibration offset to calculate the real transmit EIRP. This in most cases is the antenna gain reduced by the waveguide loss between the measurement point and the antenna.

The device monitors also a minimum and maximum threshold to raise an low or high EIRP fault and optionally the Communication Fault of the power measurement device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the level calculation.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	**description**
pwrMeasId	The message ID for the output power reading.
config.faultId	The message ID to detect and communication fault of the measurement device
config.noPwrThreshold	This value defines when the power reading indicates that there is "no power". Helpful for equipment which returns low power reading even if there is no output
config.noPwrEirp	This is the return value of `measuredEirp` if a "no power" condition is detected. Default is `-99.99` which was the behavior until driver version 1.06

Limit Checking

At the primary device page, min and max threshold values may be defined. These values apply to the measured EIRP. If the measured EIRP exceeds the limit values a low or high EIRP fault is generated. The fault gets cleared if the EIRP value returns inside the limit window or a "no power" condition is detected.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.faultId	TEXT	SAVE SETUP	StringRange
config.noPwrThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.noPwrEirp	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
offset	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.meas	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
internal.flt	BOOLEAN	NOPRESET	BooleanRange (true false)
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
noPower	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Meas.Device Comm.)

9.317 Level-AddRed

The Level-AddRed logical device provides a simple mechanism to monitor transmit power values in EIRP. The Level-AddRed device uses a fixed calibration offset and a dynamic difference to reflect losses in e.g. a redundant waveguide switching system. Expect of these location dependend offsets it work like the Level-Add device.

The Level-AddRed device monitors a configurable parameter for power measurement and adds a calibration offset and a difference to calculate the real transmit EIRP. This in most cases is the antenna gain reduced by the waveguide loss between the measurement point and the antenna and the difference regarding the location of the redundant chain. It monitors also a switch position to find which additional offset (difference) has to be added.

The device monitors also a minimum and maximum threshold to raise an low or high EIRP fault and optionally the Communication Fault of the power measurement device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the level calculation.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	description
pwrMeasId	The message ID for the output power reading.
config.faultId	The message ID to detect and communication fault of the measurement device
config.noPwrThreshold	This value defines when the power reading indicates that there is "no power". Helpful for equipment which returns low power reading even if there is no output
config.noPwrEirp	This is the return value of `measuredEirp` if a "no power" condition is detected. Default is `-99.99` which was the behavior until driver version 1.06
locationId	Parameter to determine the current redundancy switch position.
diff.1...12	Constant difference to add regarding the redundancy position.
loc.1...12	Value of the redundancy switch position.

Limit Checking

At the primary device page, min and max threshold values may be defined. These values apply to the measured EIRP. If the measured EIRP exceeds the limit values a low or high EIRP fault is

generated. The fault gets cleared if the EIRP value returns inside the limit window or a "no power" condition is detected.

Variables defined by this device driver

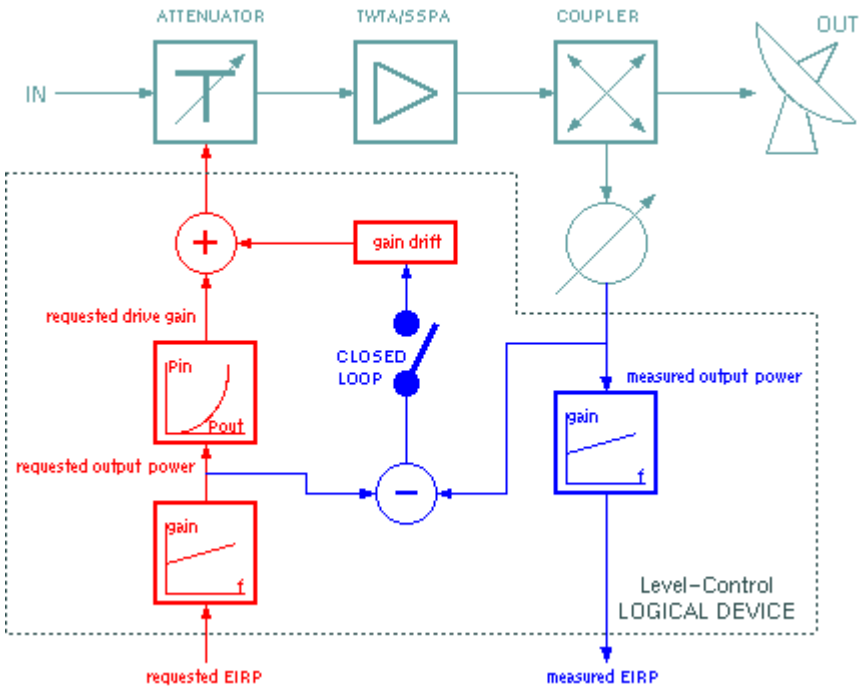
name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.faultId	TEXT	SAVE SETUP	StringRange
config.noPwrThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.noPwrEirp	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.locationId	TEXT	SAVE SETUP	StringRange
config.diff.1	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.2	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.3	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.4	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.diff.5	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.6	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.7	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.8	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.9	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.10	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.11	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.diff.12	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.loc.1	TEXT	SAVE SETUP	StringRange
config.loc.2	TEXT	SAVE SETUP	StringRange
config.loc.3	TEXT	SAVE SETUP	StringRange
config.loc.4	TEXT	SAVE SETUP	StringRange
config.loc.5	TEXT	SAVE SETUP	StringRange
config.loc.6	TEXT	SAVE SETUP	StringRange
config.loc.7	TEXT	SAVE SETUP	StringRange
config.loc.8	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.loc.9	TEXT	SAVE SETUP	StringRange
config.loc.10	TEXT	SAVE SETUP	StringRange
config.loc.11	TEXT	SAVE SETUP	StringRange
config.loc.12	TEXT	SAVE SETUP	StringRange
offset	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.meas	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
internal.flt	BOOLEAN	NOPRESET	BooleanRange (true false)
internal.location	TEXT	NOPRESET	StringRange
internal.diff	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
noPower	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Meas.Device Comm.)

9.318 Level-Control

The Level-Control logical device provides a gain setting linearizer for TWT amplifiers as well as a level control loop mode. The figure below explains how the Level-Control device works. The RF devices controlled by the Level-Control are shown in cadet blue, the level setting appears in red and the level measurement is marked blue.



When the operator sets an EIRP value, the Level-Control device first computes the amplifier output power which is required for this EIRP. The conversion is done according to a user supplied calibration table, it takes the actual signal frequency into account as well as the signal path.

The requested amplifier output power then is translated into the requested drive level. For TWT amplifiers the Pin/Pout compression curve is used for this. The Level-Control device is able to measure this characteristic by itself, however, the measurement points of this curve may be entered manually, too.

Before setting the drive value at the device used for this (usually a modem, upconverter or the gain setting parameter of the amplifier itself), the Level-Control adds a parameter called "gain drift" to the setting.

After the signal passed the amplifier, the Level-Control reads the measured output power. With amplifiers providing a sufficiently precise output power reading, this value can be used, with other amplifiers a separate power meter will be required. The measured output power gets translated to EIRP by means of the same calibration table mentioned above.

In close loop mode, the Level-Control device takes the difference between the requested output power and the measured power on a regular basis, sets the gain drift parameter to this value and adjusts the drive level accordingly. The closed loop level adjustment, even if enabled by the operator, automatically is suppressed in one of the following cases:

- The requested power is below a user specified threshold, the meter readings are expected to be noisy or even invalid in this range.
- The requested power is above a user specified threshold, near the amplifier saturation the closed loop adjustment could overdrive the amplifier.
- The measured power differs from the requested value less than a user specified

threshold. This prevents the loop from oscillating with 0.1 dB around the programmed value.
- The measured power differs from the requested value more than 3 dB. It is likely that something is going completely wrong in this case (output power is completely missing, e.g.). It is not desirable, that the drive level rises to the maximum in such a case. Instead, the Level-Control shows a fault and freezes the drive level.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational parameters, meter readings and status displays.
- --- This page shows the parameter required for the calibration of the amplifier's Pin/Pout characteristic.
- --- The faults page.
  - **High Drive Limit** : The output power reaches the high limit
  - **Low Drive Limit** : The output power reaches the low limit
  - **Gain Deviation** : Warns that there is not the expected output power, maybe because of a fault in the transmit chain.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Control device to the devices it monitors and controls.

parameter	***description***
rfOn1Id	
rfOn2Id	
rfOn3Id	You define the message IDs of up to three RF-ON switches in the transmit chain. For instance, the carrier can be switched on/off at the modulator, upconverter and the HPA.
ampStateId	Enter the "state" parameter of a TWTA here if applicable. The Level-Control monitors the state and suppresses the closed loop adjustment if the amplifier is not in "BEAM ON" state. Leave this empty if you're using a SSPA.
levelSetId	This is the message ID for the drive setting.
levelReadId	This is the message ID for the output power reading.

parameter	***description***
gain1Id	This is the message ID for the first of two invariant gain settings, the Level-Control issues with a RF-ON command. This may be used to ensure certain level settings in the transmit chain. Leave this empty, if there is no gain setting to be initialized at RF-ON.
gain2Id	This is the message ID for a second invariant gain setting. Leave this empty, if not applicable
gain1	The value to be sent to "gain1Id"
gain2	The value to be sent to "gain2Id"
txFrequencyId	The message ID of the transmit frequency, usually the upconverter tx.frequency setting. The Level-Control device listens to this parameter to perform the amplifier output power vs. EIRP calculations based on the actual frequency.
txSwitch1Id	The message ID of the first of two switches in the transmit signal path which affect the EIRP calibration. Usually here the ID of the polarization switch is entered.
txSwitch2Id	The message ID of a second switch in the transmit signal path.
redPosId	In cases where the Level-Control device controls a redundant HPA in a N:1 configuration and deltas shall be applied to the input/output calibration of the device depending on the transmit chain this one is actually replacing, then the parameter ID of the "location" parameter of the Protection-Switch-N-To-1 device must be entered here. In all other cases the redPosId parameter must stay empty. For details see the paragraph "N:1 redundant configuration" below.
redEirpDeltas	A comma separated list of delta values to be included to the EIRP calculation if the above parameter "redPosId" is set. For details see the paragraph "N:1 redundant configuration" below.
redDriveDeltas	A comma separated list of delta values to be included to the amplifier drive calculation if the above parameter "redPosId" is set. For details see the paragraph "N:1 redundant configuration" below.
maxDisplay	The EIRP value to be displayed at the right edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.

parameter	***description***
minDisplay	The EIRP value to be displayed at the left edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.

Calibration parameters

At the "Calibration" page of the device window there are a number of parameters which mark out the basic performance of the device. They are explained below.

- *parameter --- description*
- gain drift --- The gain drift is a corrective offset which the Level-Control device adds to each drive setting in order to compensate the long term gain drift with respect to the Pin/Pout calibration. In closed loop mode the gain drift is calculated and updated from the target/actual comparison. You may overwrite this value by entering another value, this value will work as a new starting point for the closed loop control.
- calibration state --- During the Pin/Pout calibration process this field shows what's going on. You start the calibration process by entering the text "start calibration" here (mall letters in lower case, no additional spaces).
- min output power --- Enter here the lowest output power reading which is not falsified by noise. Do not set this value too low, the deviance produced by the level control due to noisy power readings mostly causes more problems than a nonlinearity of the drive setting attenuator.Closed loop mode is suppressed for power settings below this limit. During the Pin/Pout calibration power readings below this limit are discarded.
- max output power --- Here the true maximum output power of the amplifier must be entered (more precisely, the power reading achieved at the maximum amplifier output). The Level-Control device may modify this value if it detects a lower saturation point during the Pin/Pout calibration procedure.
- delay --- This parameter determines the delay between a drive setting and the next measurement sample taken for the target/actual comparison. The delay also applies for the Pin/Pout calibration.
- hysteresis --- In closed loop mode, the Level-Control device does not send a new drive value if the target/actual values differ by this value or less. In calibration mode the hysteresis value is used for the saturation point detection.
- compression curve --- The Pin/Pout transfer characteristic as a list of measurement samples. The format is "Pout *1* ,Pin *1* ,Pout *2* ,Pin *2* , ... Pout *n* ,Pin *n* ". The samples need not to be equally spaced, however the Pout values must be in ascending order. In most cases you will not enter this series of measurement manually. The Level-Control device is capable to perform this measurement by itself.

Leveling And Pin/Pout Calibration

To make the Level-Control device work properly, the level ranges must be adjusted and the Pin/Pout calibration curve must contain values which exactly represent the amplifier's characteristic. When defining the levels / level ranges the following should be considered.

- All devices in the transmit chain should be operated at their nominal input level if possible, the maximum input level must not be exceeded.
- Usually the invariant gain values are set in a way, that the amplifier may be overdriven by 2 .. 3 dB with the maximum drive setting. This ensures that the amplifier reaches it's maximum power, even if the overall gain of the transmit chain degrades.

Below the steps necessary to adjust the Level-Control device for a certain setup are described.

1. First of all, the message identifiers for all devices communicating with the Level-Control device must be set. See the paragraph "Configuration Parameters" above for details.
2. To see reasonable EIRP readings, a EIRP calibration table with at least rough values must be present. Edit the calibration table with a text editor program of your choice, the paragraph "EIRP Calibration Table" below explains the format of this file.
3. Now the invariant gain settings in the transmit chain must be set to values which meet the requirements mentioned above. If you set these values as configuration parameters at the maintenance page of this device, you can rely on the values to be set with each RF-ON you command.
4. Enter the min/max amplifier output power to the adequate fields at the "Calibration" page. The maximum value should be set to the true saturation output power, the minimum value to the lowest power reading you will rely on.
5. On the calibration page, set the parameter "delay" to 10 seconds, the "hysteresis" parameter to 0.2 dB.
6. Now make sure, that the amplifier is switched on and warmed up and the output is routed to a dummy load which is capable to admit the full amplifier power. Enter the text *start calibration* to the field "calibration state" and hit RETURN. The command must be entered exactly as shown, this is a kind of safeguard against an unintended start of the calibration procedure.

The Level-Control device now starts to measure the Pin/Pout characteristic of the amplifier. It sets the lowest possible drive value, waits 10 seconds (the value set with the "delay" parameter) an then takes a reading of the output power. If the power reading is below the minimum value, this measurement point is discarded. If considered to be OK, the measurement point is added to the power transfer characteristic. The Level-Control device increases the drive by 1 dB and starts the next measurement cycle. The measurement procedure stops if

- The maximum amplifier output power is reached.
- The maximum drive value is reached.
- The amplifier output power did not increase by at lest 0.2 dB (the hysteresis value) with respect to reading at the previous drive setting which was 1dB below. The Level-Control device assumes that the saturation point is just to be reached (This condition only applies for the upper half of the drive setting range. At the low edge noisy power readings unintentionally could trigger this.)

The ongoing measurement procedure can be watched with the messages displayed in the "calibration state" and "state" fields. At any time the measurement may be aborted by commanding a RF-OFF at the device's main settings page.

Unless aborted by the operator, the LevelControl device replaces the stored Pin/Pout curve by

the recently measured one when it finishes. Additionally the "gain drift" value is cleared and the maximum amplifier output power value is adjusted to the saturation value if applicable.

EIRP Calibration Table

The Level-Control device requires an EIRP calibration table which defines the level difference between the measured output power reading and the EIRP over the frequency. This in most cases is the antenna gain reduced by the waveguide loss between the measurement point and the antenna.

The calibration table is a plain text file, called "XXXX.eirpCal" where XXXX is the name of the Level-Control device. The file must reside in the M&C software's home directory. Below an example for such a file is shown:

```
#
# /home/mnc/EIRP-1.eirpCal
#
#        TRANSMIT 1    TRANSMIT-2
#        X     Y     X     Y
  14000.0  20.2   20.2   20.2   20.2
  14100.0  20.27  20.27  20.27  20.27
  14200.0  20.31  20.31  20.31  20.31
  14300.0  20.44  20.44  20.44  20.44
  14400.0  20.5   20.5   20.5   20.5
  14500.0  20.5   20.5   20.5   20.5
```

The file is formatted as a table of 5 columns and an arbitrary number of lines. Empty lines and lines starting with a "#" are ignored. Columns are separated by spaces.

column	***description***
1	The frequency value in MHz. The frequency values must be sorted in the order low values first.
2	The gain value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=A (OFF).
3	The gain value for the signal path indicated by SWITCH1=B (ON), SWITCH2=A(OFF).
4	The gain value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=B (ON).
5	The gain value for the signal path indicated by SWITCH1=B (ON), SWITCH2=B (ON).

Limit Checking

At the primary device page, two pairs of min / max threshold values may be defined. These

values apply to the measured EIRP, they are checked while the Level-Set device is set RF-ON. If the measured EIRP exceeds the limit values a fault is generated. The fault gets cleared if the EIRP value returns inside the limit window of id the EIRP is switched off. If the Level-Set device shall not do any limit checking, set both threshold values to 0.0.

The minEirp/maxEirp settings are included into the 'gauge' parameter which controls the appearance of a Gauge screen element which follows this parameter. There is a second parameter of the same format called 'gauge2' which includes the minEirp2/maxEirp2 limits instead.

N:1 Redundant Configuration

For a Level-Control device operating a redundant transmit chain in a N:1 redundant configuration, the Pin/Pout calibration must be corrected depending on which chain actually is replaced.

For this case the Level-Control device provides a mechanism to monitor the actual "location" setting of the Protection-Switch-N-To-1 device managing the redundancy control an to apply individual offsets to the displayed EIRP and initial drive power values for each chain.

To use this feature, fill out the configuration settings redPosId, redEirpDeltas, and redDriveDeltas as described below:

- *parameter --- description*
- redPosId --- This parameter must be set to the parameter ID of the "location" of the redundancy control device. The Level-Control device is programmed to monitor a Protection-Switch-N-To-1 device, however other implementations of redundancy control devices may work as well if the publish the selected chain with a parameter containing something like "CHAIN-01" or "CHAIN-12".
- redEirpDeltas --- This parameter must be set to a comma separated list of deltas to be included into the EIRP calculation for each individual chain. The first value in the list applies to CHAIN 01, the last value applies to the highest numbered chain. If the location parameter contains a value which cannot be parsed as a chain number (e.g. "NONE") or if a chain number is addressed which is not contained in the list, a value of 0.0 is assumed.The delta value gets added to the amplifier output power (in addition to the regular calibration value) to get the displayed EIRP value. This means, if for a particular chain there is an additional attenuation in the signal path, this attenuation has to be entered as a *negative* value.
- redDriveDeltas --- This parameter must be set to a comma separated list of deltas to be included into the drive power calculation for each individual chain. The first value in the list applies to CHAIN-01, the last value applies to the highest numbered chain. If the location parameter contains a value which cannot be parsed as a chain number (e.g. "NONE") or if a chain number is addressed which is not contained in the list, a value of 0.0 is assumed.The delta value gets added to the drive / input power (in addition to the regular calibration value) when the Level-Control sets the power value. This means, if for a particular chain there is an additional attenuation in the signal path, this attenuation has to be entered here as a *positive* value.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.rfOn1Id	TEXT	SAVE SETUP	StringRange
config.rfOn2Id	TEXT	SAVE SETUP	StringRange
config.rfOn3Id	TEXT	SAVE SETUP	StringRange
config.ampStateId	TEXT	SAVE SETUP	StringRange
config.levelSetId	TEXT	SAVE SETUP	StringRange
config.levelReadId	TEXT	SAVE SETUP	StringRange
config.gain1Id	TEXT	SAVE SETUP	StringRange
config.gain2Id	TEXT	SAVE SETUP	StringRange
config.gain1	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.gain2	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.txSwitch1Id	TEXT	SAVE SETUP	StringRange
config.txSwitch2Id	TEXT	SAVE SETUP	StringRange
config.redLocId	TEXT	SAVE SETUP	StringRange
config.redEirpDeltas	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.redDriveDeltas	TEXT	SAVE SETUP	StringRange
config.maxDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.minDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
compression	TEXT	SAVE	StringRange
calibrate	TEXT	NOPRESET	StringRange
delay	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
drift	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
minAmpPower	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
maxAmpPower	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
hysteresis	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
closedLoop	CHOICE	SAVE	EnumRange (OFF ON)
on	CHOICE	SAVE	EnumRange (OFF ON)
eirp	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
minEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
measuredPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
drive	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
loopState	TEXT	R/O	StringRange R/O
redEirpDelta	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
redDriveDelta	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
gauge	TEXT	R/O	StringRange R/O
gauge2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (High drive limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low drive limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Gain deviation)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low EIRP threshold 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High EIRP threshold 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low EIRP threshold 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (High EIRP threshold 2)

9.319 Level-Monitoring

The Level-Monitoring logical device monitors a level reading like a beacon level and raises several fault flags depending on the monitored level compared to some thresholds defined on the first device window page. Beside the level value itself the device also monitors the slope of this level, uses this to detect a falling level in critical situations quickly.

This device primarily is intended to monitor a beacon level and to make the decision for a site diversity switch. Its function is best explained by describing the parameter available on the first device window page:

- **levelthreshold** --- If the monitored level drops below this value, the device raises a "Threshold" fault. This fault by default is set to WARNING priority. It does not cause a site switch immediately but with the level below this threshold the device pays closer attention to changes of the level.
- **leveldelay** --- If the level is (constantly) below the threshold value for this time, the device raises the "Low level" fault. This fault may be configured to cause a site switch.
- **slopethreshold** --- While the level is below the threshold, the device also monitors the slope of the input signal. If there is a fast signal drop (negative slope below the given threshold) a "Slope" fault is raised. Please note that the threshold value must be entered as a negative value (e.g. -1.0 dB/sec) to achieve the desired behaviour. Due to the technical nature of a slope, the "Slope" fault will probably only be present for a single driver cycle.
- **backupthreshold** --- This second level threshold causes the "Backup" fault to be raised if the input signal drops below it. It works independently from the thresholds described above and is intended to be used as some sort of inhibit signal for a site switch: "do not switch to this antenna, the beacon signal is too bad here"

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational parameters.
- --- The device info page.
- --- The faults page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there is one single configuration parameter which must be set to interconnect the Level-Monitoring device to its signal source:

parameter	***description***
inputId	The message ID of the parameter to monitor, e.g. **BCRX.power**

Remarks

The Level-Monitoring device is not limited to the usage described above. By tweaking the parameters individual functions combined in this device may be used separately:

- By setting the level threshold to a high value, the device goes permanently to the "below threshold" mode. You may ignore the level threshold faults and use device as a simple slope limit detector. If you configure the slope threshold to a positive instead of a negative number as described above, the slope fault is raised when the input signal raises with a slope faster than the given limit.
- If you ignore the "Low level" and "Slope" faults, the device may be used as a simple 2-stage minimum level check device: If the level drops below the first threshold it raises a warning, if the level drops below a second (lower) threshold, a fault is raised.

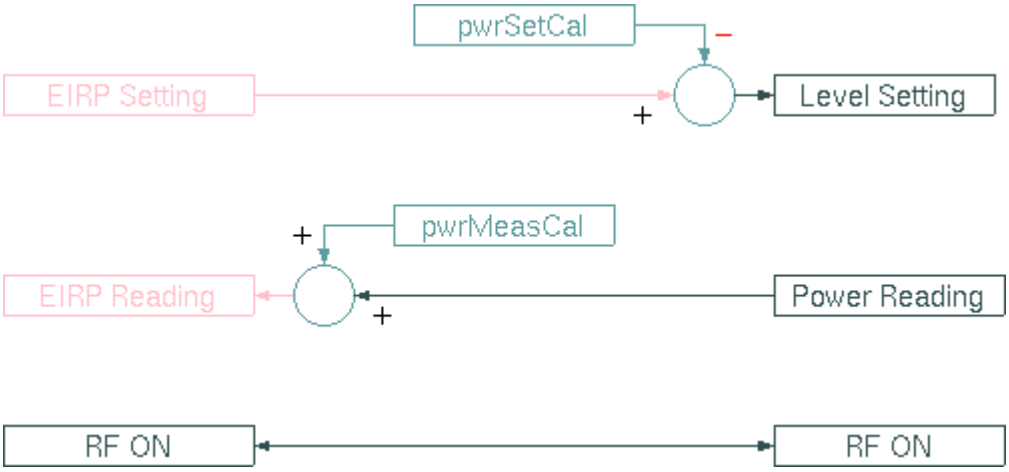
Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputId	TEXT	SAVE SETUP	StringRange
levelThreshold	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
levelDelay	INTEGER	SAVE	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
slopeThreshold	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
backupThreshold	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slope	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
lowLevelTime	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Level)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slope)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Backup)

9.320 Level-Set

The Level-Set logical device provides a simple mechanism to set (and monitor) transmit power values in EIRP. It is intended to be used with small VSAT stations using a SSPA at a fixed frequency. The Level-Set device uses fixed calibrations offsets or optionally calibration tables.



The diagram above illustrates the function of the Level-Set device. The device provides two calibrations offsets. One is used to translate an EIRP setting into a gain / output power setting to be applied at a device. The other offset translates a power reading into an EIRP value which is reported to the user. The RF-ON switch simply is routed through the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- This page shows the operational parameters.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
rfOnId	The message ID controlling the RF ON state, e.g. **KSTAR.tx.on**
pwrSetId	The message ID for the drive setting.
pwrMeasId	The message ID for the output power reading.
pwrSetCal	The calibration offset to be applied at the drive setting. The offset is defined in a way that the drive setting is calculated as:
drive-setting = EIRP(dBW) - pwrSetCal .	
The Level-Set device optionally may use calibration tables (see below), the pwrSetCal value then gets recalculated with each change of the transmit frequency or the signal path.	
pwrMeasCal	The calibration offset to be applied at the power reading. The offset is defined that the displayed EIRP is calculated as:
EIRP(dBW) = power-reading + pwrMeasCal .	
The Level-Set device optionally may use calibration tables (see below), the pwrSetCal value then gets recalculated with each change of the transmit frequency or the signal path.	

parameter	***description***
txFrequencyId	The message ID of the transmit frequency, usually the upconverter tx.frequency setting. The Level-Set device listens to this parameter to perform the amplifier output power vs. EIRP calculations based on the actual frequency.
txSwitch1Id	The message ID of the first of two switches in the transmit signal path which affect the EIRP calibration. Usually here the ID of the polarization switch is entered.
txSwitch2Id	The message ID of a second switch in the transmit signal path.
fileName	The base name of the files containing the calibration tables (see below). The name is without suffix.
maxDisplay	The EIRP value to be displayed at the right edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.
minDisplay	The EIRP value to be displayed at the left edge of a GaugeElement which uses the 'gauge' status variable provided by this device. Set maxDisplay and minDisplay both to 0.0 in order to set the gauge's display range to the complete valid range of the eirp setting.

EIRP Calibration Tables

The Level-Set device may use fixed calibration values to convert the measures amplifier power to EIRP and to convert the desired EIRP to a gain or output power setting of an arbitrary device. For applications which require a more precise power setting/display, the Level-Set device alternatively may use calibration tables. With these tables, the calibration offsets are calculated frequency dependent for up to four signal paths.

The calibration tables are plain text files, called "XXXX.eirpCal" and "XXXX.driveCal" where XXXX is the 'fileName' configuration setting. The files must reside in the M&C software's home directory. If the files exist and the Level-Set device receives a valid frequency setting, the calibration values calculated from the tables replace the pwrSetCal and pwrMeasCal values in the configuration setup.

Below an example for such a file is shown:

```
#
# /home/mnc/EIRP-1.eirpCal
#
#        TRANSMIT 1    TRANSMIT-2
#        X     Y     X     Y
  14000.0  20.2   20.2   20.2   20.2
  14100.0  20.27  20.27  20.27  20.27
  14200.0  20.31  20.31  20.31  20.31
  14300.0  20.44  20.44  20.44  20.44
  14400.0  20.5   20.5   20.5   20.5
  14500.0  20.5   20.5   20.5   20.5
```

The file is formatted as a table of 5 columns and an arbitrary number of lines. Empty lines and lines starting with a "#" are ignored. Columns are separated by spaces.

column	***description***
1	The frequency value in MHz. The frequency values must be sorted in the order low values first.
2	The calibration value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=A (OFF).
3	The calibration value for the signal path indicated by SWITCH1=B (ON), SWITCH2=A(OFF).
4	The calibration value for the signal path indicated by SWITCH1=A (OFF), SWITCH2=B (ON).
5	The calibration value for the signal path indicated by SWITCH1=B (ON), SWITCH2=B (ON).

The calibration values in the '*.eirpCal' table are defined the same way as the pwrMeasCal setup value, the values in the '*.driveCal' table are defined the same way as the pwrSetCal value.

Dynamic Change Of Calibration Settings

If the transmit frequency or the signal path changes, the Level-Set device keeps the drive setting constant and adjusts the diyplayed EIRP value according to the changed calibarion offsets.

The device provides a variable called 'adjustDrive'. If an arbitrary value gets written to this variable, the device adjusts the drive setting in order to meet the previously set EIRP value.

Limit Checking

At the primary device page, two pairs of min / max threshold values may be defined. These values apply to the measured EIRP, they are checked while the Level-Set device is set RF-ON. If

the measured EIRP exceeds the limit values a fault is generated. The fault gets cleared if the EIRP value returns inside the limit window of id the EIRP is switched off. If the Level-Set device shall not do any limit checking, set both threshold values to 0.0.

The minEirp/maxEirp settings are included into the 'gauge' parameter which controls the appearance of a Gauge screen element which follows this parameter. There is a second parameter of the same format called 'gauge2' which includes the minEirp2/maxEirp2 limits instead.

Variables defined by this device driver

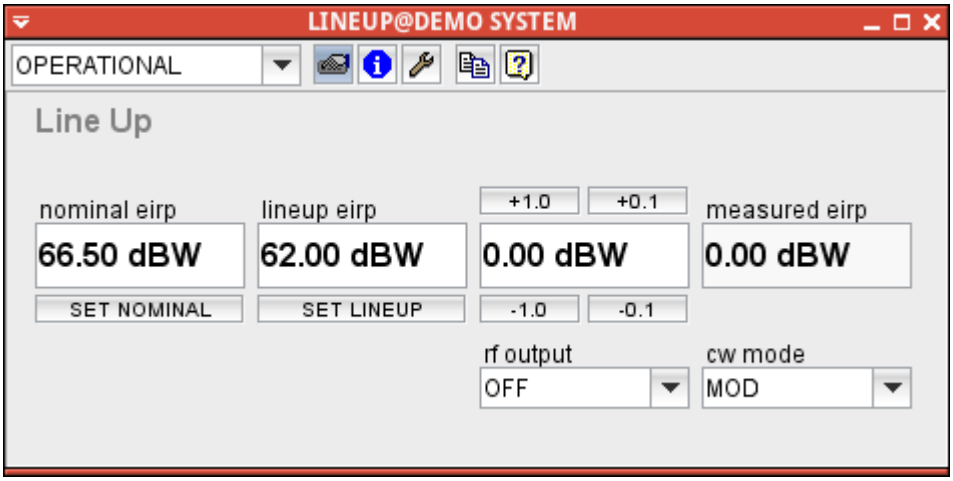
name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.rfOnId	TEXT	SAVE SETUP	StringRange
config.pwrSetId	TEXT	SAVE SETUP	StringRange
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.txSwitch1Id	TEXT	SAVE SETUP	StringRange
config.txSwitch2Id	TEXT	SAVE SETUP	StringRange
config.pwrSetCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.pwrMeasCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.fileName	TEXT	SAVE SETUP	StringRange
config.maxDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.minDisplay	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
eirp	FLOAT		DoubleRange (0.00 .. 0.00)
on	CHOICE		EnumRange (OFF ON)
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
minEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp2	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
adjustDrive	TEXT	NOPRESET	StringRange
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
gauge	TEXT	R/O	StringRange R/O
gauge2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EIRP low threshold 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EIRP high threshold 2)

9.321 LineUp-Control

The *LineUp-Control* logical device supports the operator with the carrier line-up process. While intended to be used with the TX-LineSettings-CDB device, it can also be used stand alone.

The device lets the operator control the parameters EIRP and RF-ON. The modulator can be switched between modulated and c/w output.



'nominal eirp' contains the nominals EIRP as defined in the line settings database. For stand alone operation of the *LineUp-Control* device the value may be entered manually.

'lineup eirp' contains the reduced transmit power value which is used to start the line up process. This value also is set by the *TX-LineSettings-CDB* device from the line settings database, may be entered manually as well.

The next value is the actual EIRP setting of the HPA / transmit power control loop, followed by the measured EIRP value. Controls for RF ON/OFF and the c/w setting complete the *LineUp-Control* device screen.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Brings up the device screen shown above.
- --- The device info page.
- --- The maintenance / configuration page

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
eirpSetId	The ID of the parameter which controls the EIRP. This typically may be something like 'EIRP-1.eirp'.
eirpMeasId	The ID of the parameter which reports the measured EIRP. May be something like 'EIRP-1.measuredEirp'.
rfOnId	The ID of the parameter controlling the carrier on/off state.

parameter	*description*
cwModeId	The ID of the modulator parameter which switches between modulated an c/w output.

Remarks

When used with a *TX-LineSettings-CDB* device, the latter must be configured to send the nominal EIRP value to the 'nominalEirp' parameter and the reduced EIRP value to the'lineUpEirp' parameter of this device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.eirpSetId	TEXT	SAVE SETUP	StringRange
config.eirpMeasId	TEXT	SAVE SETUP	StringRange
config.rfOnId	TEXT	SAVE SETUP	StringRange
config.cwModeId	TEXT	SAVE SETUP	StringRange
nominalEirp	FLOAT		DoubleRange (0.00 .. 0.00)
lineUpEirp	FLOAT		DoubleRange (0.00 .. 0.00)
eirp	FLOAT		DoubleRange (0.00 .. 0.00)
measEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rfOn	CHOICE		EnumRange (OFF ON)
cwMode	CHOICE		EnumRange (CD MOD)
incr	TEXT		StringRange

9.322 Link-Mirror

The Link-Mirror logical device is an intermediate workaround to control a 1:1 redundant transmit facility through a Link device. You are strongly discouraged to use this device as it removed as soon as the scheduling system has been re-worked.

There is no further online help for this type of device. Please refer to the general description of

device windows instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.frequencyId.1	TEXT	SAVE SETUP	StringRange
config.powerId.1	TEXT	SAVE SETUP	StringRange
config.onId.1	TEXT	SAVE SETUP	StringRange
config.preset1Id.1	TEXT	SAVE SETUP	StringRange
config.preset2Id.1	TEXT	SAVE SETUP	StringRange
config.preset3Id.1	TEXT	SAVE SETUP	StringRange
config.fault1Id.1	TEXT	SAVE SETUP	StringRange
config.fault2Id.1	TEXT	SAVE SETUP	StringRange
config.fault3Id.1	TEXT	SAVE SETUP	StringRange
config.frequencyId.2	TEXT	SAVE SETUP	StringRange
config.powerId.2	TEXT	SAVE SETUP	StringRange
config.onId.2	TEXT	SAVE SETUP	StringRange
config.preset1Id.2	TEXT	SAVE SETUP	StringRange
config.preset2Id.2	TEXT	SAVE SETUP	StringRange
config.preset3Id.2	TEXT	SAVE SETUP	StringRange
config.fault1Id.2	TEXT	SAVE SETUP	StringRange
config.fault2Id.2	TEXT	SAVE SETUP	StringRange
config.fault3Id.2	TEXT	SAVE SETUP	StringRange
config.switchPosId	TEXT	SAVE SETUP	StringRange
config.switchPos1	TEXT	SAVE SETUP	StringRange
frequency	FLOAT		DoubleRange (0 .. 0)

name	*type*	*flags*	*range*
power	FLOAT		DoubleRange (0 .. 0)
on	CHOICE		EnumRange (OFF ON)
preset1	GENERIC		ObjectRange generic object
preset2	GENERIC		ObjectRange generic object
preset3	GENERIC		ObjectRange generic object
fault1	TEXT	R/O	StringRange R/O
fault2	TEXT	R/O	StringRange R/O
fault3	TEXT	R/O	StringRange R/O

9.323 Link-Static

The Link-Static logical device connects the equipment used in a VLC for one end point of a *static* satellite link to the link management of the NMS software. The Link-Static device does not implement any equipment control nor does it contain a local scheduler. It simply collects the fault states of some device with a logical 'or' function and assigns the resulting state to the satellite link using these devices. The 'Link-Static' device is much easier to setup and configure than the other two link devices, it is therefore ideal for large networks where no scheduling is needed. The 'Link-Static' device may be used with static links only. If a link end point shall be used for scheduled links, you should use the Link-Two-Way or the Link device instead.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual link state
- --- The faults page.
  - **Device 1,2,3** : Summary fault of the used devices
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the Link device to the devices it monitors / controls. Unused parameters should be left empty.

parameter	*description*
mode	Selects if this Link-Static device stands for the transmit side (TX) or the receive side (RX) of a link. If set to the value RXTX, the link endpoint is used with bidirectional links only.
fault1Id	The message ID of a device fault to be monitored by the Link device.
fault2Id	The message ID of a device fault to be monitored by the Link device.
fault3Id	The message ID of a device fault to be monitored by the Link device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.mode	CHOICE	SAVE SETUP	EnumRange (RX TX RXTX)
config.fault1Id	TEXT	SAVE SETUP	StringRange
config.fault2Id	TEXT	SAVE SETUP	StringRange
config.fault3Id	TEXT	SAVE SETUP	StringRange
reset	CHOICE		EnumRange (CLEAR)
activeLink	TEXT	R/O	StringRange R/O
scheduledLinks	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Device 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Device 3)

9.324 Link-Two-Way

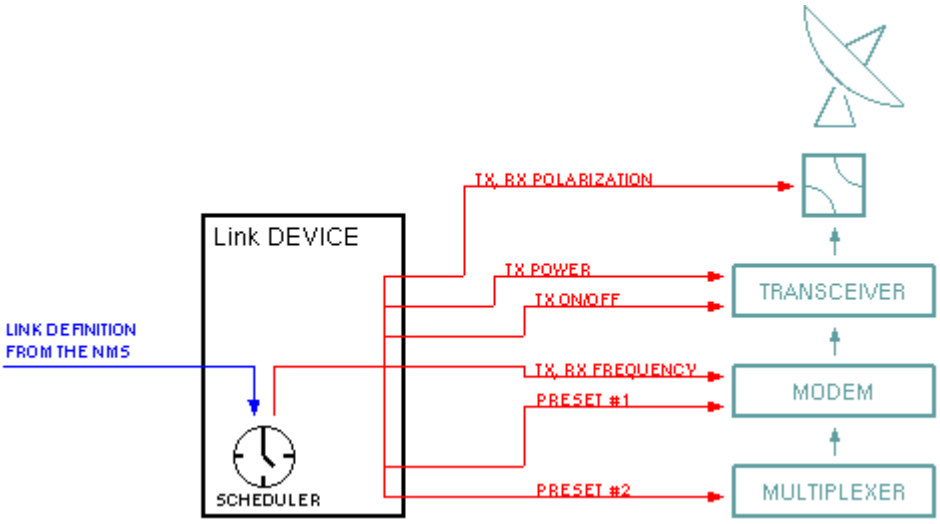
The Link-Two-Way logical device connects the equipment used in a VLC for one end point of a satellite link to the link management of the NMS software. It controls the timing when links are going up or down and does all equipment settings necessary for the satellite link. Beside this it does some sort of abstraction of the equipment settings, making the NMS capable to handle

differently equipped VLCs in the same way.

The Link-Two-Way device is used for link end points which receive and transmit at the same time. Only bidirectional links may end on a 'Link-Two-Way' device. It may be used for static and scheduled links as well. Typical applications for this link device are data or Internet link stations. There are two more variants of the link device which are adapted to other applications:

The Link device is used for link end points which either receive or transmit a signal. Only unidirectional links may end on a 'Link' device. It may be used for static and scheduled links as well. Typical applications for this are the transmit or receive facilities of a video distribution system.

The Link-Static device is a stripped down version of other two link devices. The 'Link-Static' device does not implement any equipment control nor does it contain a local scheduler. It simply collects the fault states of some device with a logical 'or' function and assigns the resulting state to the satellite link using these devices. The 'Link-Static' device is much easier to setup and configure than the other two link devices, it is therefore ideal for large networks where no scheduling is needed. The 'Link-Static' device may be used with static links only.



The figure above illustrates the function of a Link-Two-Way device used in a simple VSAT like station. As the Link-Two-Way device is widely configurable, this applies to a Video/SNG application or other configurations as well. From the NMS the Link-Two-Way device receives link definition records, data sets which define what type of carrier at which frequency shall be transmitted/received at which time. The Link-Two-Way device enlists this record into it's local scheduler. At the programmed time, it sets the RF equipment to the parameter of the requested link.

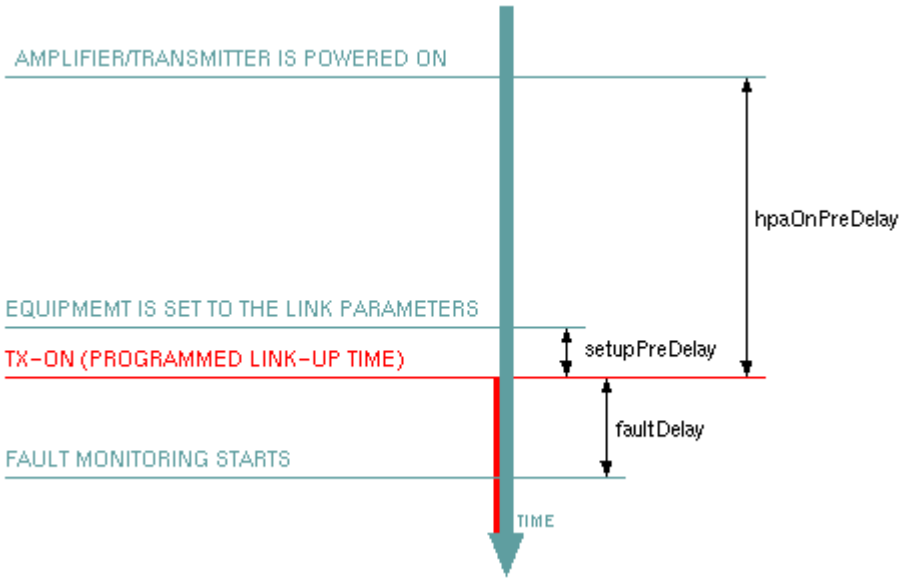
Link parameters

The link definitions record directly defines the common parameters of the satellite link like frequency, polarization or EIRP. Application specific parameters like data rate, encoding etc. are enveloped in device presets which are sent to up to three devices involved in the satellite link. Using device presets the Link-Two-Way device is capable to handle completely different

types of satellite links in the same way.

Link timing

The scheduler in the Link-Two-Way device is a little more sophisticated than the introduction above suggests. When bringing up a satellite link, the scheduler takes care of several points of time beside the effective start of the link.



Some seconds (the "setupPreDelay" configuration parameter defines this value) before the link starts, the scheduler already sets all link parameters as device presets, frequencies, polarization settings to ensure that all equipment is readily set at the programmed link-up time. At the effective start point of the link, only the tx-power is set "ON".

For applications where a station transmits only occasionally, the scheduler is able to maintain a "transmitter-on" switch which powers on the amplifier / transmitter only on demand. The "hpaOnPreDelay" configuration parameter defines how long before the start of the transmission the transmitter shall be powered on to allow a sufficient warming up period of this unit. The scheduler keeps the transmitter switched on after the link goes down, if another link if programmed within the "hpaOnPreDelay" interval.

The link device monitors the fault signals of up to three devices and reports a link OK/FAULT state to the NMS from the summary of these signals. As it is likely that when the link goes up, some devices show a fault until they lock on the signal, fault monitoring starts with some delay after tx-on was set. The "faultDelay" configuration parameter sets this time delay.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual link state

- --- The faults page.
  - **Device 1,2,3** : Summary fault of the used devices.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the Link-Two-Way device to the devices it monitors / controls. Unused parameters should be left empty.

parameter	***description***
preset1Id	The message ID to send the preset #1 to. This must be in accordance with the link type definition in the NMS.
preset2Id	The message ID to send the preset #2 to.
preset3Id	The message ID to send the preset #3 to.
txFrequencyId	The message ID to send the transmit frequency to.
txPowerId	The message ID to send the tx power value to.
txPowerCal	You may define a power calibration factor here to convert the Link EIRP into a output power setting for a modem e.g..
txOnId	The message ID to send the tx on/off command to.
txPolId	The message ID of the transmit polarization select switch.
rxPolId	The message ID of the receive polarization select switch.
txPolXlt	The transmit polarization / switch position translation table. For a wave guide switch selecting horizontal polarization at position "A" the table would be "X=A,Y=B".
rxPolXlt	Like txPolXlt, but for the receive polarization.
rxFrequencyId	The message ID to send the receive frequency to.
hpaOnId	The message ID of a switch which separately switches the power amplifier (or transceiver) on.
fault1Id	The message ID of a device fault to be monitored by the Link device.
fault2Id	The message ID of a device fault to be monitored by the Link device.
fault3Id	The message ID of a device fault to be monitored by the Link device.
faultDelay	Fault monitoring starts this time after the Link-UP (TX-ON) was commanded. This lets the equipment some time to come in lock.

parameter	*description*
hpaOnPreDelay	The power amplifier (or transceiver) gets switched on this time before the effective start of a link.
hpaOnInterval	The Link device checks in this intervals whether to switch on/off the power amplifier. This time is equivalent to the post-carriage of the amplifier after the link goes down.
setupPreDelay	The presets (containing the data rate definitions etc.) are sent to their devices this time prior to the link start time. This makes allowance for the devices to carry out the preset settings before the link really goes up.

Remarks

- The scheduler in the Link device never cuts a running transmission in order to setup the equipment for the next link before it's start point. Hence, if you program links at the NMS without a gap in between, the "setupPreDelay" will not be used. The Link device start to set the link parameters at the effective starting point of the link in this case.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.mode	CHOICE	SAVE SETUP	EnumRange (RXTX)
config.preset1Id	TEXT	SAVE SETUP	StringRange
config.preset2Id	TEXT	SAVE SETUP	StringRange
config.preset3Id	TEXT	SAVE SETUP	StringRange
config.rxFrequencyId	TEXT	SAVE SETUP	StringRange
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.powerId	TEXT	SAVE SETUP	StringRange
config.powerCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.onId	TEXT	SAVE SETUP	StringRange
config.rxPolId	TEXT	SAVE SETUP	StringRange
config.txPolId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.rxPolXlt	TEXT	SAVE SETUP	StringRange
config.txPolXlt	TEXT	SAVE SETUP	StringRange
config.hpaOnId	TEXT	SAVE SETUP	StringRange
config.fault1Id	TEXT	SAVE SETUP	StringRange
config.fault2Id	TEXT	SAVE SETUP	StringRange
config.fault3Id	TEXT	SAVE SETUP	StringRange
config.faultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hpaOnPreDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hpaOnInterval	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.setupPreDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
reset	CHOICE		EnumRange (CLEAR)
activeLink	TEXT	R/O	StringRange R/O
scheduledLinks	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Device 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Device 3)

9.325 Link

The Link logical device connects the equipment used in a VLC for one end point of a satellite link to the link management of the NMS software. It controls the timing when links are going up or down and does all equipment settings necessary for the satellite link. Beside this it does some sort of abstraction of the equipment settings, making the NMS capable to handle differently equipped VLCs in the same way.

The Link device is used for link end points which either receive or transmit a signal. Only unidirectional links may end on a 'Link' device. It may be used for static and scheduled links as well. Typical applications for this are the transmit or receive facilities of a video distribution system. There are two more variants of the link device which are adapted to other applications:

The Link-Two-Way device is used for link end points which receive and transmit at the same

time. Only bidirectional links may end on a 'Link-Two-Way' device. It may be used for static and scheduled links as well. Typical applications for this link device are data or Internet link stations.

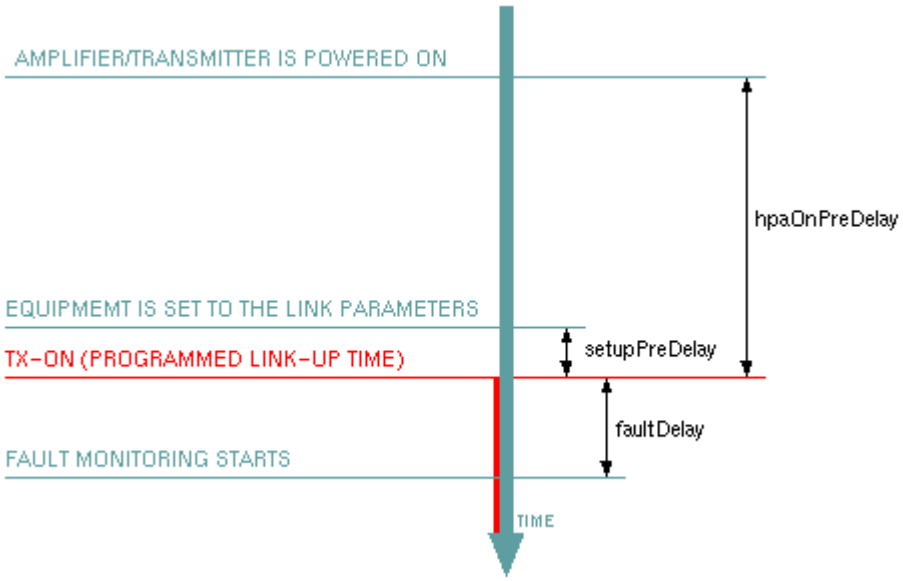
The Link-Static device is a stripped down version of other two link devices. The 'Link-Static' device does not implement any equipment control nor does it contain a local scheduler. It simply collects the fault states of some device with a logical 'or' function and assigns the resulting state to the satellite link using these devices. The 'Link-Static' device is much easier to setup and configure than the other two link devices, it is therefore ideal for large networks where no scheduling is needed. The 'Link-Static' device may be used with static links only.

Link parameters

The link definitions record directly defines the common parameters of the satellite link like frequency, polarization or EIRP. Application specific parameters like data rate, encoding etc. are enveloped in device presets which are sent to up to three devices involved in the satellite link. Using device presets the Link device is capable to handle completely different types of satellite links in the same way.

Link timing

The scheduler in the Link device is a little more sophisticated than the introduction above suggests. When bringing up a satellite link, the scheduler takes care of several points of time beside the effective start of the link.



Some seconds (the "setupPreDelay" configuration parameter defines this value) before the link starts, the scheduler already sets all link parameters as device presets, frequencies, polarization settings to ensure that all equipment is readily set at the programmed link-up time. At the effective start point of the link, only the tx-power is set "ON".

For applications where a station transmits only occasionally, the scheduler is able to maintain a "transmitter-on" switch which powers on the amplifier / transmitter only on demand. The

"hpaOnPreDelay" configuration parameter defines how long before the start of the transmission the transmitter shall be powered on to allow a sufficient warming up period of this unit. The scheduler keeps the transmitter switched on after the link goes down, if another link if programmed within the "hpaOnPreDelay" interval.

The link device monitors the fault signals of up to three devices and reports a link OK/FAULT state to the NMS from the summary of these signals. As it is likely that when the link goes up, some devices show a fault until they lock on the signal, fault monitoring starts with some delay after tx-on was set. The "faultDelay" configuration parameter sets this time delay.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual link state
- --- The faults page.
  - **Device 1,2,3** : Summary fault of the used devices
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the Link device to the devices it monitors / controls. Unused parameters should be left empty.

parameter	***description***
mode	Selects if this Link device controls the transmit side (TX) or the receive side (RX) of a link.
preset1Id	The message ID to send the preset #1 to. This must be in accordance with the link type definition in the NMS.
preset2Id	The message ID to send the preset #2 to.
preset3Id	The message ID to send the preset #3 to.
frequencyId	The message ID to send the transmit/receive frequency to.
powerId	The message ID to send the tx power value to. Should be left empty if the Link device controls receive equipment.
powerCal	You may define a power calibration factor here to convert the Link EIRP into a output power setting for a modem e.g..
onId	The message ID to send the tx on/off command to.

parameter	*description*
polId	The message ID of the polarization select switch.
polXlt	The polarization / switch position translation table. For a wave guide switch selecting horizontal polarization at position "A" the table would be "X=A,Y=B".
hpaOnId	The message ID of a switch which separately switches the power amplifier (or transceiver) on.
fault1Id	The message ID of a device fault to be monitored by the Link device.
fault2Id	The message ID of a device fault to be monitored by the Link device.
fault3Id	The message ID of a device fault to be monitored by the Link device.
faultDelay	Fault monitoring starts this time after the Link-UP (TX-ON) was commanded. This lets the equipment some time to come in lock.
hpaOnPreDelay	The power amplifier (or transceiver) gets switched on this time before the effective start of a link.
hpaOnInterval	The Link device checks in this intervals whether to switch on/off the power amplifier. This time is equivalent to the post-carriage of the amplifier after the link goes down.
setupPreDelay	The presets (containing the data rate definitions etc.) are sent to their devices this time prior to the link start time. This makes allowance for the devices to carry out the preset settings before the link really goes up.

Remarks

- The scheduler in the Link device never cuts a running transmission in order to setup the equipment for the next link before it's start point. Hence, if you program links at the NMS without a gap in between, the "setupPreDelay" will not be used. The Link device start to set the link parameters at the effective starting point of the link in this case.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.mode	CHOICE	SAVE SETUP	EnumRange (RX TX)

name	*type*	*flags*	*range*
config.preset1Id	TEXT	SAVE SETUP	StringRange
config.preset2Id	TEXT	SAVE SETUP	StringRange
config.preset3Id	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.powerId	TEXT	SAVE SETUP	StringRange
config.powerCal	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.onId	TEXT	SAVE SETUP	StringRange
config.polId	TEXT	SAVE SETUP	StringRange
config.polXlt	TEXT	SAVE SETUP	StringRange
config.hpaOnId	TEXT	SAVE SETUP	StringRange
config.fault1Id	TEXT	SAVE SETUP	StringRange
config.fault2Id	TEXT	SAVE SETUP	StringRange
config.fault3Id	TEXT	SAVE SETUP	StringRange
config.faultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hpaOnPreDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hpaOnInterval	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.setupPreDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
reset	CHOICE		EnumRange (CLEAR)
activeLink	TEXT	R/O	StringRange R/O
scheduledLinks	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Device 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Device 3)

9.326 Logger

The Logger logical device watches an arbitrary parameter and, writes a message to the event log every time the parameter value changes. The message written to the log contains a user definable text and the new parameter value.

The Logger device may be used to journalize any type of parameter, however, fluctuating values like power measurements will produce a huge amount of data. Typical applications for the Logger device are to monitor position changes of signal switches, carrier on/off states or frequency settings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Logger device with the devices that will perform the frequency setting.

parameter	***description***
messageId	Enter the message identifier here of the parameter the logger device shall listen to, e.g. "MODULATOR.tx.on".
text	Enter the message text which shall be written to the event log if the watched parameter changes. For example, "Modulator output switched" results to a log message "Modulator output switched ON" or "Modulator output switched OFF", depending of the new value of the parameter.

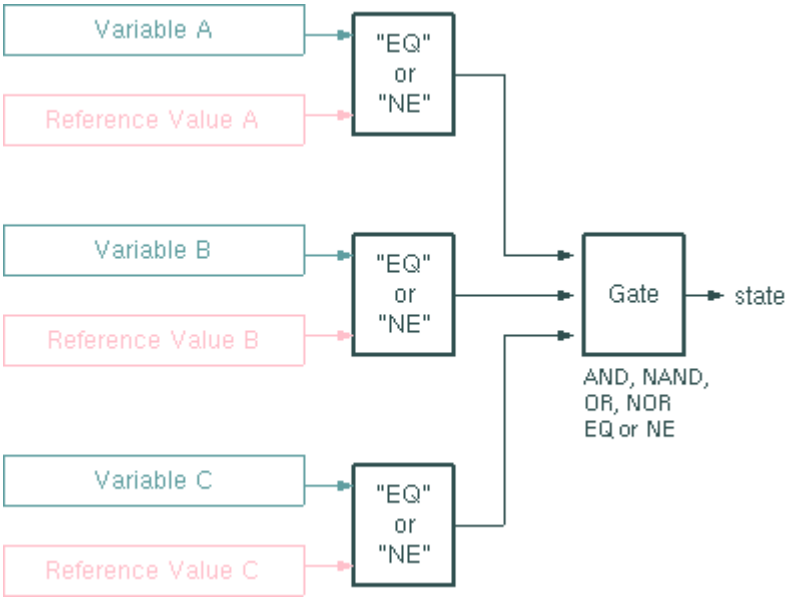
Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.messageId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.text	TEXT	SAVE SETUP	StringRange

9.327 Logical-Gate-3

The Logical-Gate-3 device provides a 3-input logical gate which can be used to merge status variables in a user programmed way. The gate may be configured to work as a AND, NAND, OR, NOR, NE (XOR) or EQ gate. It listens to two input variables and compares these to user supplied reference values. the logical result of the comparisons gets fed into the gate. The gate's output state is distributed as a boolean state variable named "state".



Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Logical-Gate device with the devices that dliver the gate's input signals.

parameter	*description*

parameter	***description***
input-A.Id	Enter the message identifier here of the parameter the gate shall listen as input-A, e.g. "SWITCH-1. position".
input-A.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-A.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
gateType	Select the type of logical gate.
input-B.Id	Enter the message identifier here of the parameter the gate shall listen as input-B, e.g. "SWITCH-2. position".
input-B.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-B.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
input-C.Id	Enter the message identifier here of the parameter the gate shall listen as input-C, e.g. "SWITCH-3. position".
input-C.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-C.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.

parameter	*description*
setFaultOnTrue	If set to YES, the device sets the device summary fault and a (hidden) fault flag **faults.01** if the gate output is true. The faults are set fully event driven without delay, therefore the **faults.01** flag cannot be configured in any way.
startupDelay	The number of cycles after startup the device waits until it produces the output. Defaults to 0 for backward compatibility. May be used to avoid glitches during the startup phase where not all input parameters are known.

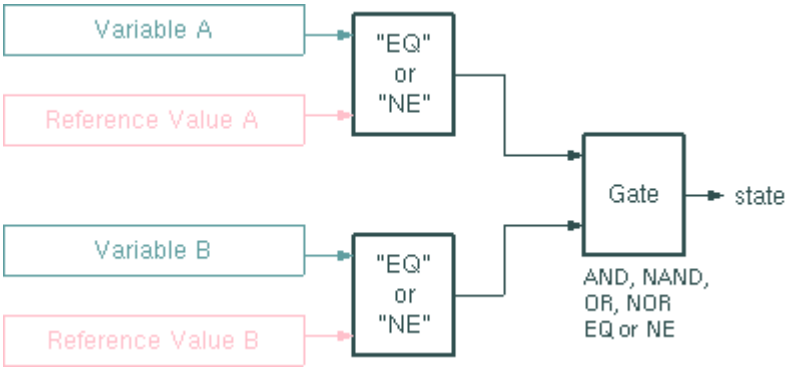
Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
info.initialized	TEXT	R/O	StringRange R/O
config.gateType	CHOICE	SAVE SETUP	EnumRange (AND NAND OR NOR EQ NE)
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-A.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-A.reference	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.input-B.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-B.reference	TEXT	SAVE SETUP	StringRange
config.input-C.Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-C.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-C.reference	TEXT	SAVE SETUP	StringRange
config.setFaultOnTrue	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.startupDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
state	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Gate output)

9.328 Logical-Gate

The Logical-Gate device provides a simple 2-input logical gate which can be used to merge status variables in a user programmed way. The gate may be configured to work as a AND, NAND, OR, NOR, NE (XOR) or EQ gate. It listens to two input variables and compares these to user supplied reference values. the logical result of the comparisons gets fed into the gate. The gate's output state is distributed as a boolean state variable named "state".



Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Logical-Gate device with the devices that deliver the gate's input signals.

parameter	***description***
input-A.Id	Enter the message identifier here of the parameter the gate shall listen as input-A, e.g. "SWITCH-1. position".
input-A.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-A.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
gateType	Select the type of logical gate.
input-B.Id	Enter the message identifier here of the parameter the gate shall listen as input-B, e.g. "SWITCH-1. position".
input-B.compare	Select **EQ** if this gate input shall become "true" if the monitored value matches the reference value defined below. **NE** reverses this, the input becomes "true" if the monitored value does not match the reference value.
input-B.reference	The reference value the monitored parameter shall be compared to. The comparison made is textual, so be careful when comparing numeric parameters. Inaccuracies of the floating point arithmetic or number formatting issues may let the monitored parameter look different than you expect.
setFaultOnTrue	If set to YES, the device sets the device summary fault and a (hidden) fault flag **faults.01** if the gate output is true. The faults are set fully event driven without delay, therefore the **faults.01** flag cannot be configured in any way.
startupDelay	The number of cycles after startup the device waits until it produces the output. Defaults to 0 for backward compatibility. May be used to avoid glitches during the startup phase where not all input parameters are known.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
info.initialized	TEXT	R/O	StringRange R/O
config.gateType	CHOICE	SAVE SETUP	EnumRange (AND NAND OR NOR EQ NE)
config.input-A.Id	TEXT	SAVE SETUP	StringRange
config.input-A.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-A.reference	TEXT	SAVE SETUP	StringRange
config.input-B.Id	TEXT	SAVE SETUP	StringRange
config.input-B.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input-B.reference	TEXT	SAVE SETUP	StringRange
config.setFaultOnTrue	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.startupDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
state	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Gate output)

9.329 Lynx-DVD5420-ASI-Distr

This page describes the device driver and the device window for the Lynx DVD5420 ASI-Distributor. One driver instance controls one 2:8 distribution module in the device. To control a complete frame with multiple distribution modules you have to configure multiple sat-nms devices.

The Lynx DVD5420 ASI-Distributor is controlled by means of a text based protocol running over

a TCP/IP socket connection. Hence, Lynx DVD5420 are to be configured in the same way than devices controlled through a TCP/IP to serial adapter / terminal server. The default port used by the device is 2306. The Port specification for a device with the IP address 10.0.0.1 would be "10.0.0.1:2306".

If multiple devices are set up to control the modules in one rack frame, the devices all must be assigned to the same port / device thread.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- All parameters of the device are shown on this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DVD5420 ASI-Distributor.

parameter	***description***
address	This value addresses the module slot within the 10 inch rack frame. You may specify the complete address as described in the Lynx protocol documentation. For a single rack frame it is sufficient to set the slot number (slot numbering starts with 1 beside the power supply).
user	The user name to log on to the device. The factory default of the DVD5420 ASI-Distributor is "user".
password	The password to log on to the device. The factory default of the DVD5420 ASI-Distributor is "lynx$ctrl".
readBackDelay	This parameter defines the time (milliseconds), the driver waits at least before it reads back a parameter which just has been changed.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwtype	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.mctype	TEXT	R/O	StringRange R/O
info.mcrevision	TEXT	R/O	StringRange R/O
info.psu1	TEXT	R/O	StringRange R/O
info.psu2	TEXT	R/O	StringRange R/O
address	TEXT	SAVE	StringRange
config.user	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
out.1.select	CHOICE		EnumRange (1 2)
out.1.reclock	CHOICE		EnumRange (OFF ON)
out.1.invert	CHOICE		EnumRange (OFF ON)
out.2.select	CHOICE		EnumRange (1 2)
out.2.reclock	CHOICE		EnumRange (OFF ON)
out.2.invert	CHOICE		EnumRange (OFF ON)
out.3.select	CHOICE		EnumRange (1 2)
out.3.reclock	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.3.invert	CHOICE		EnumRange (OFF ON)
out.4.select	CHOICE		EnumRange (1 2)
out.4.reclock	CHOICE		EnumRange (OFF ON)
out.4.invert	CHOICE		EnumRange (OFF ON)
out.5.select	CHOICE		EnumRange (1 2)
out.5.reclock	CHOICE		EnumRange (OFF ON)
out.5.invert	CHOICE		EnumRange (OFF ON)
out.6.select	CHOICE		EnumRange (1 2)
out.6.reclock	CHOICE		EnumRange (OFF ON)
out.6.invert	CHOICE		EnumRange (OFF ON)
out.7.select	CHOICE		EnumRange (1 2)
out.7.reclock	CHOICE		EnumRange (OFF ON)
out.7.invert	CHOICE		EnumRange (OFF ON)
out.8.select	CHOICE		EnumRange (1 2)
out.8.reclock	CHOICE		EnumRange (OFF ON)
out.8.invert	CHOICE		EnumRange (OFF ON)
in.1.state	TEXT	R/O	StringRange R/O
in.2.state	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
internal.auth	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input 1 missing)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input 2 missing)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input 1 invalid)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input 2 invalid)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU 1)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Authentication)

9.330 Lynx-DVD5820-ASI-Distr

This page describes the device driver and the device window for the Lynx DVD5820 ASI-Distributor. One driver instance controls one distribution module in the device with two inputs and a group of 4 output per input (which are handled together). To control a complete frame with multiple distribution modules you have to configure multiple sat-nms devices.

The Lynx DVD5480 ASI-Distributor is controlled by means of a text based protocol running over a TCP/IP socket connection. Hence, Lynx DVD5480 are to be configured in the same way than devices controlled through a TCP/IP to serial adapter / terminal server. The default port used by the device is 2306. The Port specification for a device with the IP address 10.0.0.1 would be "10.0.0.1:2306".

If multiple devices are set up to control the modules in one rack frame, the devices all must be assigned to the same port / device thread.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- All parameters of the device are shown on this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DVD5420 ASI-Distributor.

parameter	*description*
slot	This value addresses the module slot within the 10 inch rack frame. You may specify the complete address as described in the Lynx protocol documentation. For a single rack frame it is sufficient to set the slot number (slot numbering starts with 1 beside the power supply).
out.1.select	Assign an Input to the first output.

parameter	*description*
out.2.select	Assign an Input to the second output.
in.1.reclock	Enabled/Disable reclocking on first input
in.2.reclock	Enabled/Disable reclocking on second input
user	The user name to log on to the device. The factory default of the DVD5420 ASI-Distributor is "user".
password	The password to log on to the device. The factory default of the DVD5420 ASI-Distributor is "lynx$ctrl".
readBackDelay	This parameter defines the time (milliseconds), the driver waits at least before it reads back a parameter which just has been changed.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwtype	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.slot	TEXT	SAVE SETUP	StringRange
config.user	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.out.1.select	CHOICE	SETUP	EnumRange (Input 1 Input 2)
config.out.2.select	CHOICE	SETUP	EnumRange (Input 1 Input 2)
config.in.1.reclock	CHOICE	SETUP	EnumRange (OFF ON)
config.in.2.reclock	CHOICE	SETUP	EnumRange (OFF ON)
displayName	TEXT	R/O	StringRange R/O
statusColor	TEXT	R/O	StringRange R/O
in.1.state	TEXT	R/O	StringRange R/O
in.2.state	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
internal.auth	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input 1 missing)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input 2 missing)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input 1 invalid)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input 2 invalid)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Authentication)

9.331 Lynx-DVD5820-Main

This page describes the device driver and the device window for the Lynx DVD5820 Chassis Monitoring Device. It shows the state of the power supplies of a DVD5820 chassis.

The Lynx DVD5480 is controlled by means of a text based protocol running over a TCP/IP socket connection. Hence, Lynx DVD5480 are to be configured in the same way than devices controlled through a TCP/IP to serial adapter / terminal server. The default port used by the device is 2306. The Port specification for a device with the IP address 10.0.0.1 would be "10.0.0.1:2306".

If multiple devices are set up to control the modules in one rack frame, the devices all must be assigned to the same port / device thread.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- All parameters of the device are shown on this page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DVD5420 ASI-Distributor.

parameter	***description***
slot	This value addresses the module slot within the 10 inch rack frame. You may specify the complete address as described in the Lynx protocol documentation. For a single rack frame it is sufficient to set the slot number (slot numbering starts with 1 beside the power supply).
user	The user name to log on to the device. The factory default of the DVD5420 ASI-Distributor is "user".
password	The password to log on to the device. The factory default of the DVD5420 ASI-Distributor is "lynx$ctrl".
readBackDelay	This parameter defines the time (milliseconds), the driver waits at least before it reads back a parameter which just has been changed.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.mctype	TEXT	R/O	StringRange R/O
info.mcrevision	TEXT	R/O	StringRange R/O
info.psu1	TEXT	R/O	StringRange R/O
info.psu2	TEXT	R/O	StringRange R/O
config.user	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
reset	TEXT	NOPRESET	StringRange
internal.auth	TEXT	R/O	StringRange R/O
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Authentication)

9.332 MCL-MT-TWTA-CSP

This page describes the device driver and the device window for the MCL MT3300, MT3400, MT4000 and MT4400 power amplifiers.

Please Note: This drivers supports the CSP protocol. For SABus protocol use MCL-MT-TWTA.device

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 49
lowLimit	The low output power limit. A value of 0 disables the limit check
highLimit	The high output power limit. A value of 0 disables the limit check
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.

Serial interface settings

You have to configure the the serial interface on the front panel of the HPA. If you want to use RS232 you have to use the "Computer" interface not the "Remote" interface.

Additionally you have to set the jumpers behind the front panel to enable RS232. For details please refer to the user manual of the HPA.

Hint: For testing RS232 communication you can use pin 6(rx), 9(tx) and 5(ground) of J5 Computer interface if the jumper is set to RS485. If set the jumper for RS232, you can use the documented pins of J5.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.filamentElapsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.beamElapsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.transmitCounter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 32.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 32.0)
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.psTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
red.switch	CHOICE		EnumRange (ANTENNA LOAD)
red.mode	CHOICE		EnumRange (AUTO MANUAL)
reset	TEXT	NOPRESET	StringRange
remote	TEXT	NOPRESET	StringRange
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF low alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF high alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Waveguide ARC)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Uncalibrated)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Auto filament off)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HV over Volt)
faults.09	ALARM	R/O	AlarmFlagRange R/O (HV under Volt)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Reflectec power fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HV prim. current)
faults.12	ALARM	R/O	AlarmFlagRange R/O (User interlock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis interlock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Tube temp. switch)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tube overdrive alarm)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Helix run current)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Power supply temp.)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Waveguide pressure)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF high fault)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Helix surge current)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Filament under current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ARC test)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Exciter)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Tube temp. fault)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Blower)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF switch failed)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AC Low Line)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Linearizer Comm)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Beam control)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Mating interlock)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RF cover interlock)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Tube overdrive fault)
faults.35	ALARM	R/O	AlarmFlagRange R/O (PPM not ready)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Reflectec power alarm)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Sat CCA comm)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Tube temp. alarm)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Summary alarm)
faults.40	ALARM	R/O	AlarmFlagRange R/O (RF Int. CCA comm)

name	*type*	*flags*	*range*
faults.41	ALARM	R/O	AlarmFlagRange R/O (PS Int. CCA comm)
faults.42	ALARM	R/O	AlarmFlagRange R/O (SWOV & User Int. CCA comm)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Circulator Alarm)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Forward Power Uncal)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Reflected Power Uncal)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Drive Power Uncal)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Attenuator Uncal)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Fiber)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Upconverter Fault)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Beam Control Fault)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Fault Reset Lockout Fault)
faults.52	ALARM	R/O	AlarmFlagRange R/O (AC Low Line Fault)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Waveguide Arc 2)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Arc 2 Test Failed)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Circulator Fault)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Blower Failed Fault)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.333 MCL-MT-TWTA

This page describes the device driver and the device window for the MCL MT3300, MT3400, MT4000 and MT4400 power amplifiers. Low Power TWT amplifiers. This driver was tested with 750W MT4430

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 49
lowLimit	The low output power limit. A value of 0 disables the limit check
highLimit	The high output power limit. A value of 0 disables the limit check
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .

Serial interface settings

You have to configure the the serial interface on the front panel of the HPA. If you want to use RS232 you have to use the "Computer" interface not the "Remote" interface.

Additionally you have to set the jumpers behind the front panel to enable RS232. For details please refer to the user manual of the HPA.

Hint: For testing RS232 communication you can use pin 6(rx), 9(tx) and 5(ground) of J5 Computer interface if the jumper is set to RS485. If set the jumper for RS232, you can use the documented pins of J5.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.filamentElapsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.beamElapsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.transmitCounter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)

name	*type*	*flags*	*range*
tx.gain	FLOAT		DoubleRange (0.0 .. 32.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 32.0)
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 60.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.text	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.byte	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block1.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block1.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block1.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block1.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block1.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block2.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.block2.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.block3.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (RF low (alarm))
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF high (alarm))
faults.05	ALARM	R/O	AlarmFlagRange R/O (Waveguide ARC)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Uncalibrated)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Auto filament off)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HV over Volt)
faults.09	ALARM	R/O	AlarmFlagRange R/O (HV under Volt)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RF reflectec power)
faults.11	ALARM	R/O	AlarmFlagRange R/O (HV prim. current)
faults.12	ALARM	R/O	AlarmFlagRange R/O (User interlock)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chassis interlock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Tube temp. switch)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tube overdrive)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Helix run current)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Power supply temp.)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Waveguide pressure)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF high)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Helix surge current)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Filament under current)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ARC test)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Exciter)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Tube temp. analog)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Blower)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF switch failed)
faults.27	ALARM	R/O	AlarmFlagRange R/O (AC Low Line)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Linearizer Comm)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Beam control)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Standby)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

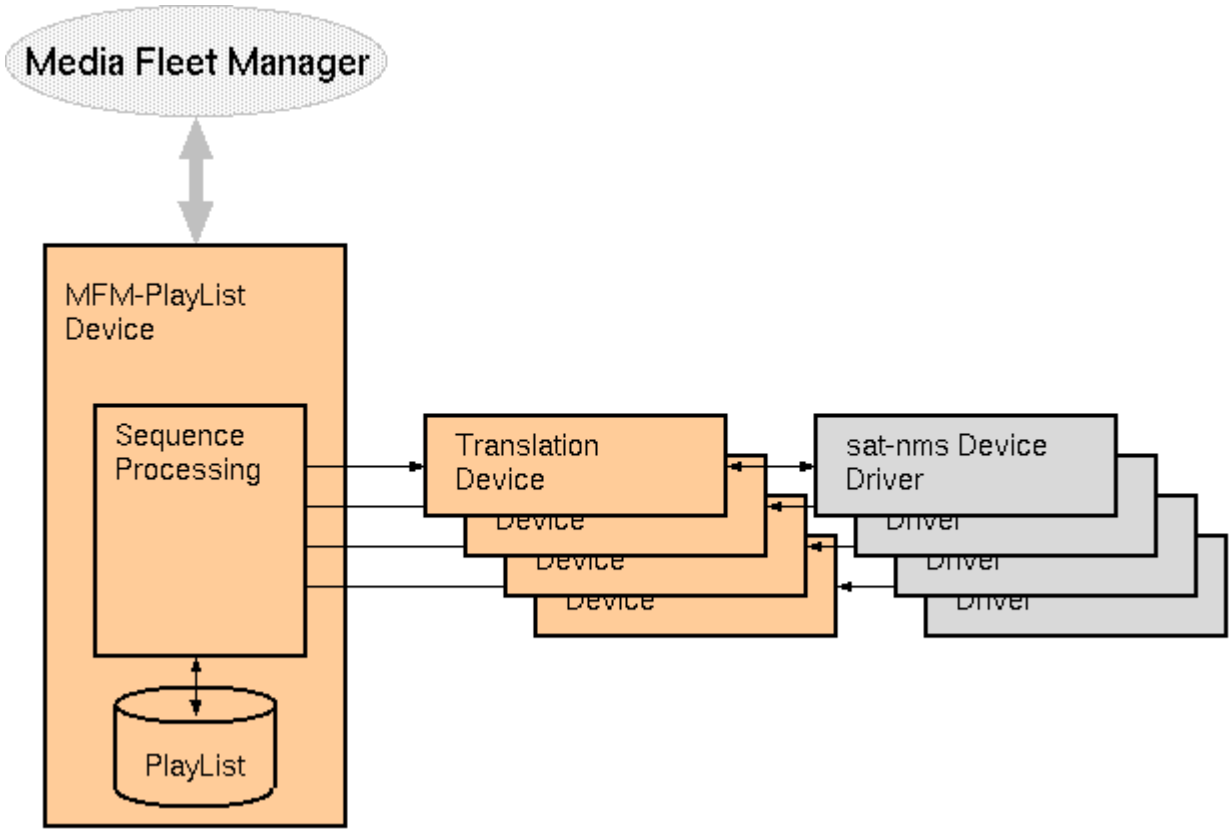
9.334 MFM-IpDecoder

The MFM-IpDecoder logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager.The MFM-IpDecoder is a so called translation device which interconnects the MFM-PlayList device to a sat-nms IP decoder device driver.

Internally the Media Fleet Manager codes all equipment settings in binary enumerations (for choice parameters) and scaled integers for numeric values. The MFM-IpDecoder translation device takes these settings and translates them into the parameter representation expected by the sat-nms device driver. For this, it inspects each parameter's range definition and matches the settings against the MFM encoding of the parameters.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General status.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
aliasName	The alias name of the device reported to the Media Fleet Manager
checkDelay	After receiving a set of parameters from the MFM-Playlist and setting these parameters at the device, the MFM-IpDecoder device waits the time set here until it checks if all parameters have been set properly.
onAirFaultDelay	After going on air, the device suppresses faults for the time configured here. This helps to avoid receiver lock faults/warnings for the first seconds after the signal is switched on.

parameter	*description*
decDevice	The device name of the IP decoder. This is used to get the device's summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
transportTypeId	The message ID of the parameter controlling the transport type (UDP/RTP).
streamingModeId	The message ID of the parameter controlling the streaming mode (UNICAST/MULTICAST).
srcIpAddressId	The message ID of the decoder parameter controlling the source IP address.
srcPortNoId	The message ID of the decoder parameter controlling the source IP port number .
encryptModeId	The message ID of the decoder parameter controlling the encryption mode (BISS-1/BISS-E).
session1WordId	The message ID of the decoder parameter setting the BISS-1 session word.
sessionEWordId	The message ID of the decoder parameter setting the BISS-E session word.
rasKeyId	The message ID of the decoder parameter setting the RAS encryption key.
streamingOnId	The message ID of a parameter switching actually "streaming on". A device driver may use this parameter to suppress missing input data faults until the stream is actually started.

All message ID settings above may be left empty. The MFM-IpDecoder device will not control a parameter with an empty ID setting and it will not check if it is set properly.

Session word and encryption key parameters are not read back and compared to the commanded values, as decoders normally do not support a readout of these values.

Remarks

- The device does not provide individual alarm flags. It reports a summary fault state which resembles the state of the decoder configured with the 'decDevice' configuration setting.
- Setting the device to the FAULT-SUPPRESSED mode will suppress alarms or warnings of the decoder which would normally lead to ONAIR-ALARM or ONAIR-WARNING alerts at the Media Fleet Manager. FAULT-SUPPRESSED does not suppress the communication fault check the MFM-PlayList device does before configuring the device nor does it suppress the check of correct parameter setting at the end of the configuration process.
- Setting the device OUT-OF-SERVICE switches it to a simulation mode in which it does not

send any command to the decoder and reports to the MFM-PlayList 'everything OK' all the time.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.aliasName	TEXT	SAVE SETUP	StringRange
config.checkDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.onAirFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.decDevice	TEXT	SAVE SETUP	StringRange
config.transportTypeId	TEXT	SAVE SETUP	StringRange
config.streamingModeId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcPortNoId	TEXT	SAVE SETUP	StringRange
config.encryptModeId	TEXT	SAVE SETUP	StringRange
config.session1WordId	TEXT	SAVE SETUP	StringRange
config.sessionEWordId	TEXT	SAVE SETUP	StringRange
config.injectedIDId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.rasKeyId	TEXT	SAVE SETUP	StringRange
config.streamingOnId	TEXT	SAVE SETUP	StringRange
set	TEXT		StringRange
check	CHOICE	R/O	EnumRange R/O (IDLE PENDING SUCCESS FAILED)
errorCode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
onAir	CHOICE		EnumRange (OFFAIR ONAIR)
releaseCheckFault	TEXT		StringRange

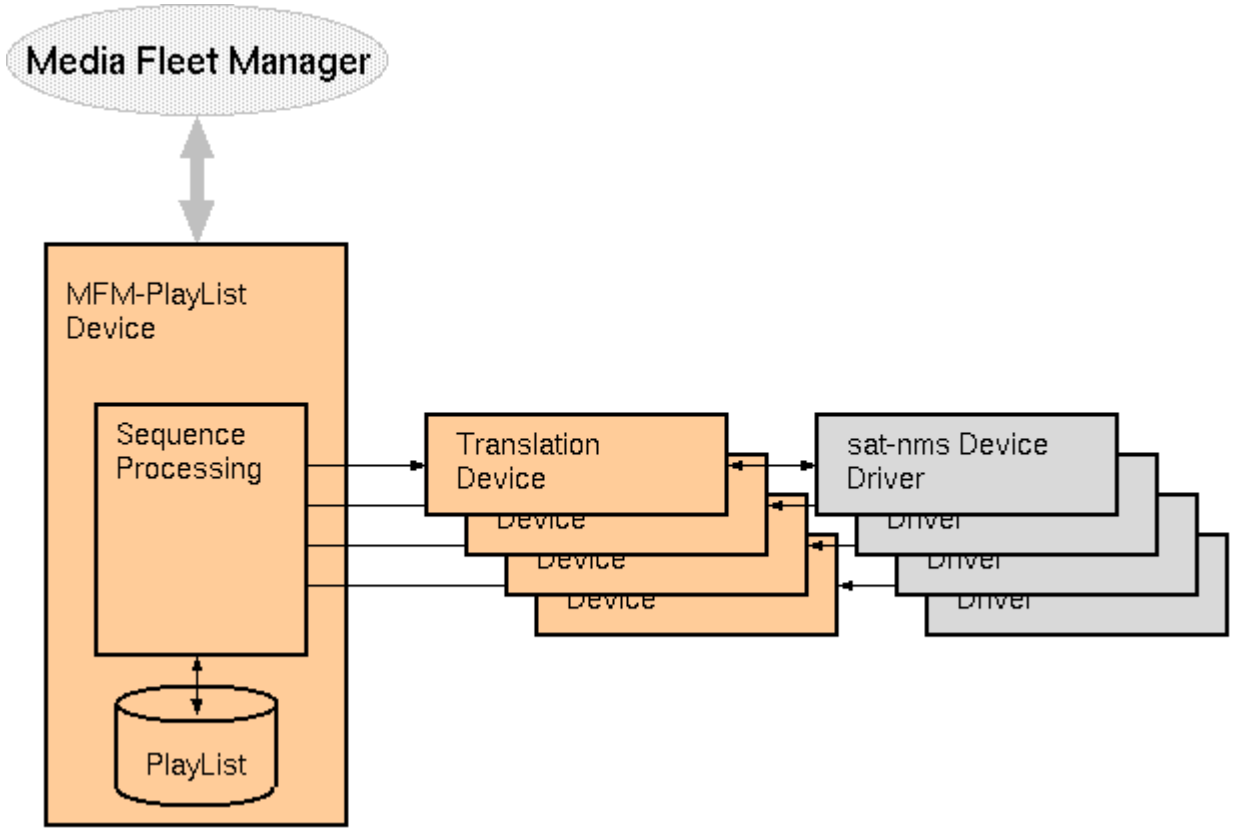
9.335 MFM-IpEncoder

The MFM-IpEncoder logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager.The MFM-IpEncoder is a so called translation device which interconnects the MFM-PlayList device to the sat-nms device driver of an IP encoder device.

Internally the Media Fleet Manager codes all equipment settings in binary enumerations (for choice parameters) and scaled integers for numeric values. The MFM-IpEncoder translation device takes these settings and translates them into the parameter representation expected by the sat-nms device drivers. For this, it inspects each parameter's range definition and matches the settings against the MFM encoding of the parameters.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General status.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
aliasName	The alias name of the device reported to the Media Fleet Manager.
checkDelay	After receiving a set of parameters from the MFM-Playlist and setting these parameters at the IRD, the MFM-IpEncoder device waits the time set here until it checks if all parameters have been set properly.
onAirFaultDelay	After going on air, the device suppresses faults for the time configured here. This helps to avoid receiver lock faults/warnings for the first seconds after the signal is switched on.
encDevice	The device name of the encoder. This is used to get the device summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
bitrateId	The message ID of the parameter controlling the transport stream bitrate.
ipBitrateId	The message ID of the parameter controlling the maximum available ip bitrate for the stream.
transportTypeId	The message ID of the parameter controlling the transport type (UDP/RTP).
streamingModeId	The message ID of the parameter controlling the streaming mode (UNICAST/MULTICAST).
dstIpAddressId	The message ID of the decoder parameter controlling the destination IP address.
dstPortNoId	The message ID of the decoder parameter controlling the destination IP port number .
packetSizeId	The message ID of the parameter controlling the packet size (188/204)
encryptModeId	The message ID of the encoder parameter controlling the encryption mode (BISS-1/BISS-E).

parameter	*description*
session1WordId	The message ID of the encoder parameter setting the BISS-1 session word.
sessionEWordId	The message ID of the encoder parameter setting the BISS-E session word.
injectedIDId	The message ID of the encoder parameter setting the BISS injected ID.
rasKeyId	The message ID of the encoder parameter setting the RAS encryption key.
audioRateId	The message ID of the encoder parameter setting the first audio channel bitrate.
audioRate2Id	The message ID of the encoder parameter setting the second audio channel bitrate.
audio2EnableId	The message ID of the encoder parameter enabling the second audio channel.
videoRateId	The message ID of the encoder parameter setting the video bitrate.
streamingOnId	The message ID of a parameter switching actually "streaming on".

All message ID settings above may be left empty. The MFM-IpEncoder device will not control a parameter with an empty ID setting and it will not check if it is set properly.

Session word and encryption key parameters are not read back and compared to the commanded values, as some devices do not support a readout of these values.

Video profiles

Encoder devices are handling encoding profile parameters in different ways, depending on vendor and / or encoder type. For this reason there is no direct translation possible from the MFM specific parameters selecting an encoding profile and the corresponding encoder parameters. Instead, the MFM-IpEncoder device calls a sat-nms device preset to configure these parameters.

To make this work, you have to store device presets for each particular encoder type you are using and the video encoding profiles used in your MFM equipment presets. The names of the presets called by the MFM-IpEncoder device follow a fixed scheme, resembling the combination of the "video coding standard" and "coding format" MFM parameters.

This is the list of all presets names recognized by the MFM-IpEncoder device:

```
MPEG2-MP@ML-420     MPEG4-MP@ML-420
MPEG2-MP@LL-420     MPEG4-MP@LL-420
MPEG2-HP@ML-422     MPEG4-HP@ML-422
MPEG2-HP@LL-422     MPEG4-HP@LL-422
MPEG2-SP@LL-420     MPEG4-SP@LL-420
MPEG2-SP@ML-420     MPEG4-SP@ML-420
MPEG2-MP@H14-420    MPEG4-MP@H14-420
MPEG2-MP@HL-420     MPEG4-MP@HL-420
MPEG2-HP@H14-422    MPEG4-HP@H14-422
MPEG2-HP@HL-422     MPEG4-HP@HL-422
MPEG2-HP@HL-420     MPEG4-HP@HL-420
MPEG2-HP@H14-420    MPEG4-HP@H14-420
MPEG2-HP@ML-420     MPEG4-HP@ML-420
```

It is not necessary to create all of these presets, only these ones addressed by MFM transmission presets are required.

If the presets are created by the "preset store" function in the device window, they should be edited afterwards to remove all parameters from the preset which are set explicitly by the MFM-IpEncoder device. If such a parameter remains in the preset, it may overwrite the value actually commanded by the MFM-IpEncoder device. There is no fixed order in which preset contained parameters and explicitly commanded ones are processed by the encoder device driver.

Remarks

- The device does not provide individual alarm flags. It reports a summary fault state which resembles the summary state of the device configured with the 'encDevice', configuration setting.
- Setting the device to the FAULT-SUPPRESSED mode will suppress alarms or warnings of the attached devices which would normally lead to ONAIR-ALARM or ONAIR-WARNING alerts at the Media Fleet Manager. FAULT-SUPPRESSED does not suppress the communication fault checks the MFM-PlayList device does before configuring the device nor does it suppress the check of correct parameter setting at the end of the configuration process.
- Setting the device OUT-OF-SERVICE switches it to a simulation mode in which it does not send commands to any device and reports to the MFM-PlayList 'everything OK' all the time.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.aliasName	TEXT	SAVE SETUP	StringRange
config.checkDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.onAirFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.encDevice	TEXT	SAVE SETUP	StringRange
config.bitrateId	TEXT	SAVE SETUP	StringRange
config.ipBitrateId	TEXT	SAVE SETUP	StringRange
config.transportTypeId	TEXT	SAVE SETUP	StringRange
config.streamingModeId	TEXT	SAVE SETUP	StringRange
config.dstIpAddressId	TEXT	SAVE SETUP	StringRange
config.dstPortNoId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange
config.encryptModeId	TEXT	SAVE SETUP	StringRange
config.session1WordId	TEXT	SAVE SETUP	StringRange
config.sessionEWordId	TEXT	SAVE SETUP	StringRange
config.injectedIDId	TEXT	SAVE SETUP	StringRange
config.rasKeyId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audioRate2Id	TEXT	SAVE SETUP	StringRange
config.audio2EnableId	TEXT	SAVE SETUP	StringRange
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.streamingOnId	TEXT	SAVE SETUP	StringRange
set	TEXT		StringRange
check	CHOICE	R/O	EnumRange R/O (IDLE PENDING SUCCESS FAILED)
errorCode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
onAir	CHOICE		EnumRange (OFFAIR ONAIR)
releaseCheckFault	TEXT		StringRange

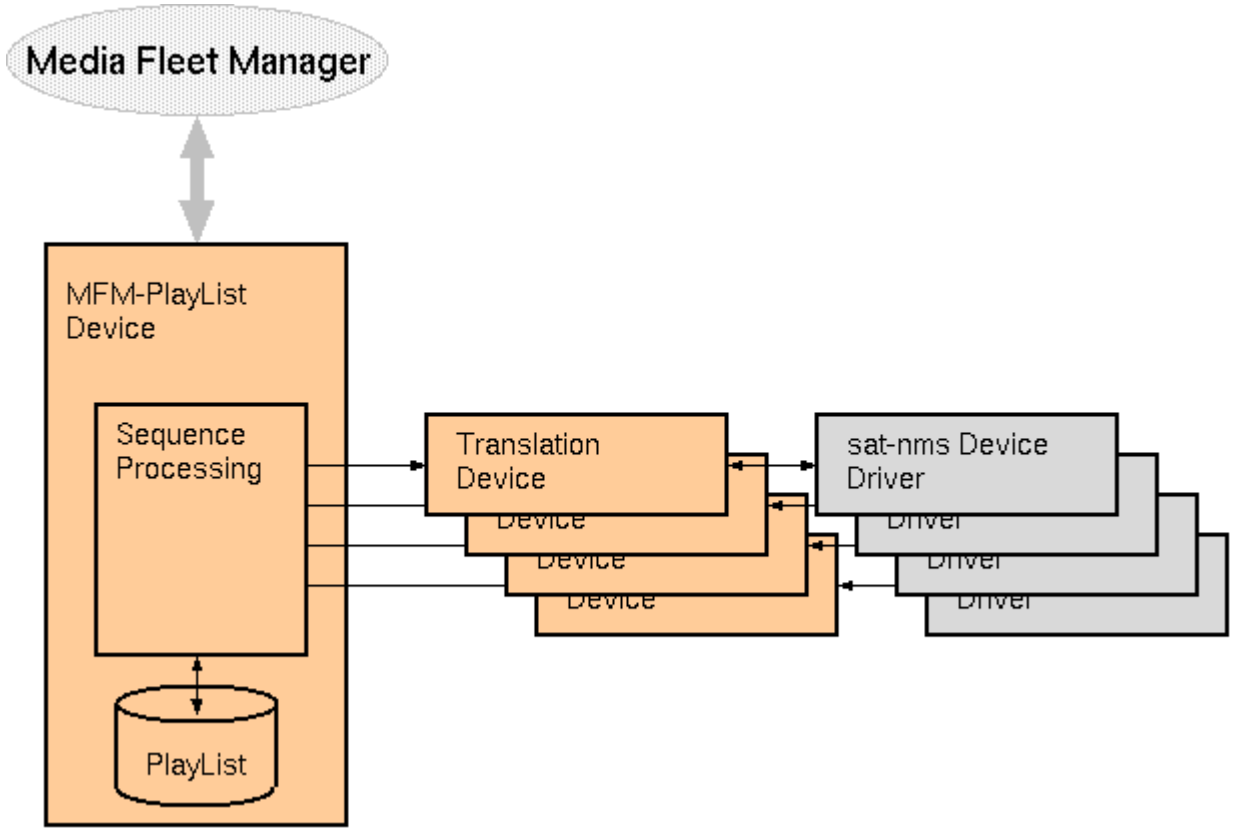
9.336 MFM-Ird

The MFM-Ird logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager.The MFM-Ird is a so called translation device which interconnects the MFM-PlayList device to a sat-nms IRD device driver.

Internally the Media Fleet Manager codes all equipment settings in binary enumerations (for choice parameters) and scaled integers for numeric values. The MFM-Ird translation device takes these settings and translates them into the parameter representation expected by the sat-nms device driver. For this, it inspects each parameter's range definition and matches the settings against the MFM encoding of the parameters.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General status.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
aliasName	The alias name of the device reported to the Media Fleet Manager
checkDelay	After receiving a set of parameters from the MFM-Playlist and setting these parameters at the IRD, the MFM-Ird device waits the time set here until it checks if all parameters have been set properly.
onAirFaultDelay	After going on air, the device suppresses faults for the time configured here. This helps to avoid receiver lock faults/warnings for the first seconds after the signal is switched on.
ignoreConfigFault	Setting this to 'YES' makes the device tolerate configuration faults at the IRD. Configuring the station proceeds even if the IRD cannot be configured. This mode is intended for IRDs in mobile SNGs where the IRD is not permanently switched on. All IRD faults are reported to the MFM as WARNING in this mode. Setting this parameter to 'NO' lets the device treat the IRD in normal way.
lockFaultMode	Selects how to treat IRD lock faults (one of the faults detected through the lockFault1Id .. lockFault3Id settings. 'OFF' ignores any faults detected this way, 'WARNING' sets the device state to (at least) WARNING if one of the lock faults is active, 'FAULT' sets the device state to FAULT in such a case. Lock faults are recognized only while the device id ONAIR and the 'onAirFaultDelay' has passed. This gives the opportunity to show an unused (unlocked) IRD device as OK in the station summary state by masking all input signal depending faults to OFF at the device. While a transmission is on air, the "lock faults" will be monitored separately and signalled to the operator.
irdDevice	The device name of the IRD. This is used to get the IRD's summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
lockFault1Id	The message ID the first IRD lock fault to monitor. Lock faults are monitored only while the device is ONAIR, faults are recognized even if masked to OFF or INFO at the IRD device itself.

parameter	***description***
lockFault2Id	The message ID the second IRD lock fault to monitor. Lock faults are monitored only while the device is ONAIR, faults are recognized even if masked to OFF or INFO at the IRD device itself.
lockFault3Id	The message ID the third IRD lock fault to monitor. Lock faults are monitored only while the device is ONAIR, faults are recognized even if masked to OFF or INFO at the IRD device itself.
frequencyId	The message ID of the parameter controlling the receive frequency. The frequency is commanded in terms of true satellite frequency, so either the IRD resp. IRD driver itself has to do the LO calculations.
polarizationId	The message ID of the parameter controlling the receive polarization plane.
symbolRateId	The message ID of the IRD parameter controlling the symbol rate.
dvbModeId	The message ID of the IRD parameter controlling the dvb mode (DVB-S/DVBS2).
modulationId	The message ID of the IRD parameter controlling the modulation type (QPSK, 8PSK, etc).
fecId	The message ID of the IRD parameter controlling the FEC.
pilotsId	The message ID of the IRD parameter switching DVB-S2 pilots on/off.
rollOffId	The message ID of the IRD parameter controlling the roll off factor.
encryptModeId	The message ID of the IRD parameter controlling the encryption mode (BISS-1/BISS-E).
session1WordId	The message ID of the IRD parameter setting the BISS-1 session word.
sessionEWordId	The message ID of the IRD parameter setting the BISS-E session word.
rasKeyId	The message ID of the IRD parameter setting the RAS encryption key.

All message ID settings above may be left empty. The MFM-Ird device will not control a parameter with an empty ID setting and it will not check if it is set properly. As many IRDs work best if parameters like modulation or FEC are kept in AUTO mode, it may be a good choice not to set the message IDs for these parameter with such a device.

Session word and encryption key parameters are not read back and compared to the commanded values, as IRDs normally do not support a readout of these values.

Remarks

- The device does not provide individual alarm flags. It reports a summary fault state which

resembles the state of the IRD configured with the 'irdDevice' configuration setting.

- Setting the device to the FAULT-SUPPRESSED mode will suppress alarms or warnings of the IRD which would normally lead to ONAIR-ALARM or ONAIR-WARNING alerts at the Media Fleet Manager. FAULT-SUPPRESSED does not suppress the communication fault check the MFM-PlayList device does before configuring the device nor does it suppress the check of correct parameter setting at the end of the configuration process.
- Setting the device OUT-OF-SERVICE switches it to a simulation mode in which it does not send any command to the IRD and reports to the MFM-PlayList 'everything OK' all the time.
- If the IRD does not provide separate parameters for DVB mode, modulation type end FEC but uses a combined MODCOD Setting for this, you may want to use the Modcod-Splitter device to control these parameters.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.aliasName	TEXT	SAVE SETUP	StringRange
config.checkDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.onAirFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.ignoreConfigFault	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.lockFaultMode	CHOICE	SAVE SETUP	EnumRange (OFF WARNING FAULT)
config.irdDevice	TEXT	SAVE SETUP	StringRange
config.lockFault1Id	TEXT	SAVE SETUP	StringRange
config.lockFault2Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.lockFault3Id	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.encryptModeId	TEXT	SAVE SETUP	StringRange
config.session1WordId	TEXT	SAVE SETUP	StringRange
config.sessionEWordId	TEXT	SAVE SETUP	StringRange
config.injectedIDId	TEXT	SAVE SETUP	StringRange
config.rasKeyId	TEXT	SAVE SETUP	StringRange
set	TEXT		StringRange
check	CHOICE	R/O	EnumRange R/O (IDLE PENDING SUCCESS FAILED)
errorCode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
onAir	CHOICE		EnumRange (OFFAIR ONAIR)

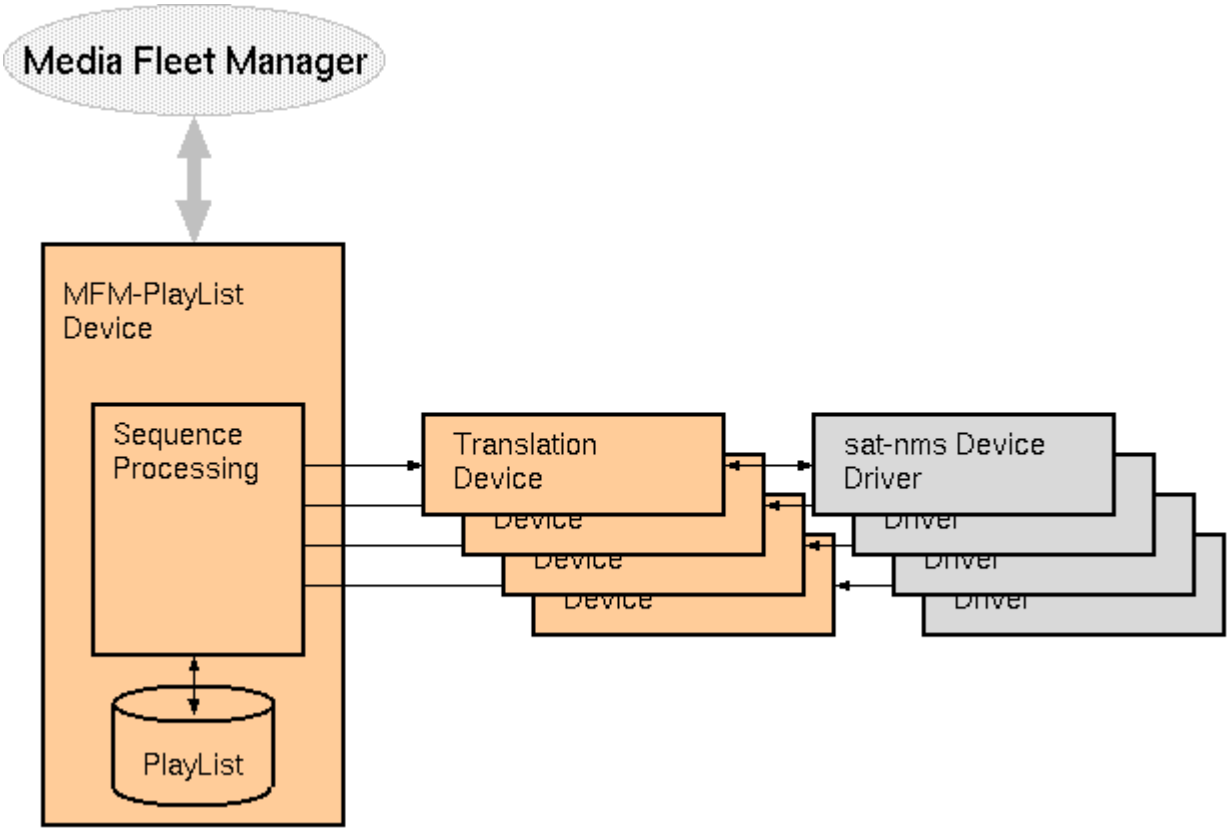
name	*type*	*flags*	*range*
releaseCheckFault	TEXT		StringRange

9.337 MFM-PlayList

The MFM-PlayList logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager.The MFM-PlayList is the central device of this framework, it provides the background processing control for scheduled transmissions and it manages the communication between Media Fleet Manager and the sat-nms VLC.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device usage

The first page of the device window shows the list of transmissions scheduled for this device. To see details for a certain transmission from the list, double-click to the list entry. Once an entry is selected, the window shows the detail parameters of this entry and also updates them automatically.

If satellite selection is configured as 'MANUAL', the window shows at it's bottom the prompt if the antenna must be moved to another satellite. Sequence processing of the actual playlist entry pauses until the operator has pointed the antenna, clicked 'DONE' and then confirmed that the antenna is pointed properly.

Right above the playlist widget, the window shows two buttons "START" and "ABORT". These button give a local operator limited control over the transmission sequence.

'START' forces the selected transmission to start immediately now. This is possible for transmissions in the states 'SCHEDULED' and 'HOTPHASE'. Please note that starting a transmission this way ahead of time bypasses all resource availability checks and may lead either to double transmissions on the satellite or to a premature end of another transmission done from this station.

'ABORT' prematurely switches a running transmission OFF-AIR. This is safely possible at any time. Aborted transmissions are signalled as 'MANUAL-LINE-DOWN' to the Media Fleet Manager, they stay in the playlist until their regular end time.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- Shows the list of scheduled transmissions
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
preconfiguration	Enables / disabled preconfiguration. With preconfiguration devices are configured already with beginning of the 'hotphase'. without preconfiguration device configuration is done exactly with the start time of a transmission, what in fact delays the effective start of transmission for a couple of seconds.
pointing	When set to 'AUTO', the MFM-PlayList device commands the name of the satellite which shall be used for the transmission directly to an antenna controller. This requires an antenna controller which is capable to this. With 'MANUAL', the MFM-PlayList device prompts the operator to point the antenna to the correct satellite and to confirm the pointing before the transmission actually will be started. Setting this parameter to 'MANUAL' will clear the 'satelliteId' and 'antennaPolId' fields.
polset	When set to 'ANTENNA', the MFM-PlayList device assumes the polarization plane for a transmission has to be done by turning the feed to the proper polarization. Again, this can be performed automatically by a command to the antenna controller or manually by the operator, depending on the 'pointing' setting above.
homeSatellite	The name of the 'home satellite', which is the one used to establish the control connection between MFM and the sat-nms station. In 'AUTO' mode, this satellite is actually commanded to the ACU after a transmission has completed. In 'MANUAL' mode, the operator is prompted to do this.
homeSatellitePol	The polarization plane used to establish the control connection to the MFM.
hotPhaseDuration	The duration of the 'hot phase', in seconds before the planned start of a transmission.
donePhaseDuration	The duration of the 'done phase', in seconds after the end of a transmission.
satelliteId	The message ID of the ACU parameter which causes the antenna controller to point the antenna to a certain satellite (only required in 'AUTO' mode).
antennaPolId	The message ID of the ACU parameter which switches the antenna's polarization plane (only effective in 'AUTO' mode with 'polset' set to 'ANTENNA').

parameter	***description***
antennaPolSense	Selects if the polarization plane commanded to the antenna refers to the transmit or to the receive side. Also sets the 'sense' of the polarization plane prompted to the operator in 'MANUAL' mode.
pointedId	The message ID of the parameter the ACU uses to confirm that the antenna is properly pointed.
courseId	The message ID of the parameter telling the course of the antenna.
latitudeId	The message ID of the parameter telling the antenna's geodetic latitude.
longitudeId	The message ID of the parameter telling the antenna's geodetic longitude.
longitudeDir	Switches the sign interpretation of the longitude values. 'WEST' means positive values count westward, 'EAST' means positive values count eastward.
sealevelId	The message ID of the parameter telling the antenna's altitude.

Settings in vlc.properties

There are a number of parameters of the MFM-PlayList device which are set in the vlc.properties file. These parameters normally should not be changed, they are listed here with their default values for completeness:

```
# The MFM task numbers

MFM.playList.taskNo=1
MFM.stationDevices.taskNo=2
MFM.systemState.taskNo=3

# The communication timeout (secs)

MFM.timeout=30

# The sat-nms listen port for connections from the MFM

MFM.portNumber=5005

# With this set to 'false' the software no longer waits
# after commanding a satellite name to an ACU until the ACU
# states 'pointed'. Only for testing purposes.

MFM.playList.waitUntilAntennaPointed=true

# Setting this true ignores communication faults of the
# attached physical devices. Only for testing purposes.

MFM.playList.ignoreCommFaults=false
```

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.initialized	BOOLEAN	R/O	BooleanRange R/O (true false)
info.connected	BOOLEAN	R/O	BooleanRange R/O (true false)
config.preconfiguration	CHOICE	SAVE SETUP	EnumRange (DISABLED ENABLED)
config.pointing	CHOICE	SAVE SETUP	EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
config.polset	CHOICE	SAVE SETUP	EnumRange (ANTENNA SWITCH)
config.homeSatellite	TEXT	SAVE SETUP	StringRange
config.homeSatellitePol	CHOICE	SAVE SETUP	EnumRange (HORIZONTAL VERTICAL LHCP RHCP)
config.hotPhaseDuration	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.donePhaseDuration	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.satelliteId	TEXT	SAVE SETUP	StringRange
config.antennaPolId	TEXT	SAVE SETUP	StringRange
config.antennaPolSense	CHOICE	SAVE SETUP	EnumRange (TX-POL RX-POL)
config.pointedId	TEXT	SAVE SETUP	StringRange
config.courseId	TEXT	SAVE SETUP	StringRange
config.latitudeId	TEXT	SAVE SETUP	StringRange
config.longitudeId	TEXT	SAVE SETUP	StringRange
config.longitudeDir	CHOICE	SAVE SETUP	EnumRange (WEST EAST)
config.sealevelId	TEXT	SAVE SETUP	StringRange
playlist	CHOICE		EnumRange (???)
operatorMessage	TEXT	R/O	StringRange R/O
antennaPointed	TEXT		StringRange
promptPending	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
debug	TEXT		StringRange
start	TEXT		StringRange
startEnabled	TEXT	R/O	StringRange R/O
abort	TEXT		StringRange
abortEnabled	TEXT	R/O	StringRange R/O
plAudioRate	TEXT	R/O	StringRange R/O
plAudioRate2	TEXT	R/O	StringRange R/O
plBitRate	TEXT	R/O	StringRange R/O
plSymbolRate	TEXT	R/O	StringRange R/O
plCodingFormat	TEXT	R/O	StringRange R/O
plConfigState	TEXT	R/O	StringRange R/O
plDestination	TEXT	R/O	StringRange R/O
plDvbEncryptionMode	TEXT	R/O	StringRange R/O
plDvbS2FrameSize	TEXT	R/O	StringRange R/O
plDvbS2PilotInsertion	TEXT	R/O	StringRange R/O
plEirp	TEXT	R/O	StringRange R/O
plErrorCode	TEXT	R/O	StringRange R/O
plFec	TEXT	R/O	StringRange R/O
plIPDestination	TEXT	R/O	StringRange R/O
plIPQualityOfService	TEXT	R/O	StringRange R/O
plIPSourceAddress	TEXT	R/O	StringRange R/O
plIPStreamingMode	TEXT	R/O	StringRange R/O
plId	TEXT	R/O	StringRange R/O
plDevices	TEXT	R/O	StringRange R/O
plInjectedID	TEXT	R/O	StringRange R/O
plInterComEirp	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
plInterComRxFreq	TEXT	R/O	StringRange R/O
plInterComTxFreq	TEXT	R/O	StringRange R/O
plIpBitRate	TEXT	R/O	StringRange R/O
plModulation	TEXT	R/O	StringRange R/O
plModulationStd	TEXT	R/O	StringRange R/O
plPacketSize	TEXT	R/O	StringRange R/O
plRASKey	TEXT	R/O	StringRange R/O
plRemark	TEXT	R/O	StringRange R/O
plRollOffFactor	TEXT	R/O	StringRange R/O
plRxFreq	TEXT	R/O	StringRange R/O
plRxPol	TEXT	R/O	StringRange R/O
plSatName	TEXT	R/O	StringRange R/O
plSessionWord	TEXT	R/O	StringRange R/O
plSkyWanChannelID	TEXT	R/O	StringRange R/O
plSkyWanSigDemodBoardID	TEXT	R/O	StringRange R/O
plSlotName	TEXT	R/O	StringRange R/O
plSource	TEXT	R/O	StringRange R/O
plStartTime	TEXT	R/O	StringRange R/O
plStatus	TEXT	R/O	StringRange R/O
plStopTime	TEXT	R/O	StringRange R/O
plTitle	TEXT	R/O	StringRange R/O
plTransportType	TEXT	R/O	StringRange R/O
plTxFreq	TEXT	R/O	StringRange R/O
plTxPol	TEXT	R/O	StringRange R/O
plVideoCodingStandard	TEXT	R/O	StringRange R/O
plVideoRate	TEXT	R/O	StringRange R/O
hotphase	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
alarm	BOOLEAN	R/O	BooleanRange R/O (true false)
onair	BOOLEAN	R/O	BooleanRange R/O (true false)
warning	BOOLEAN	R/O	BooleanRange R/O (true false)
actualSatellite	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (MFM communication)

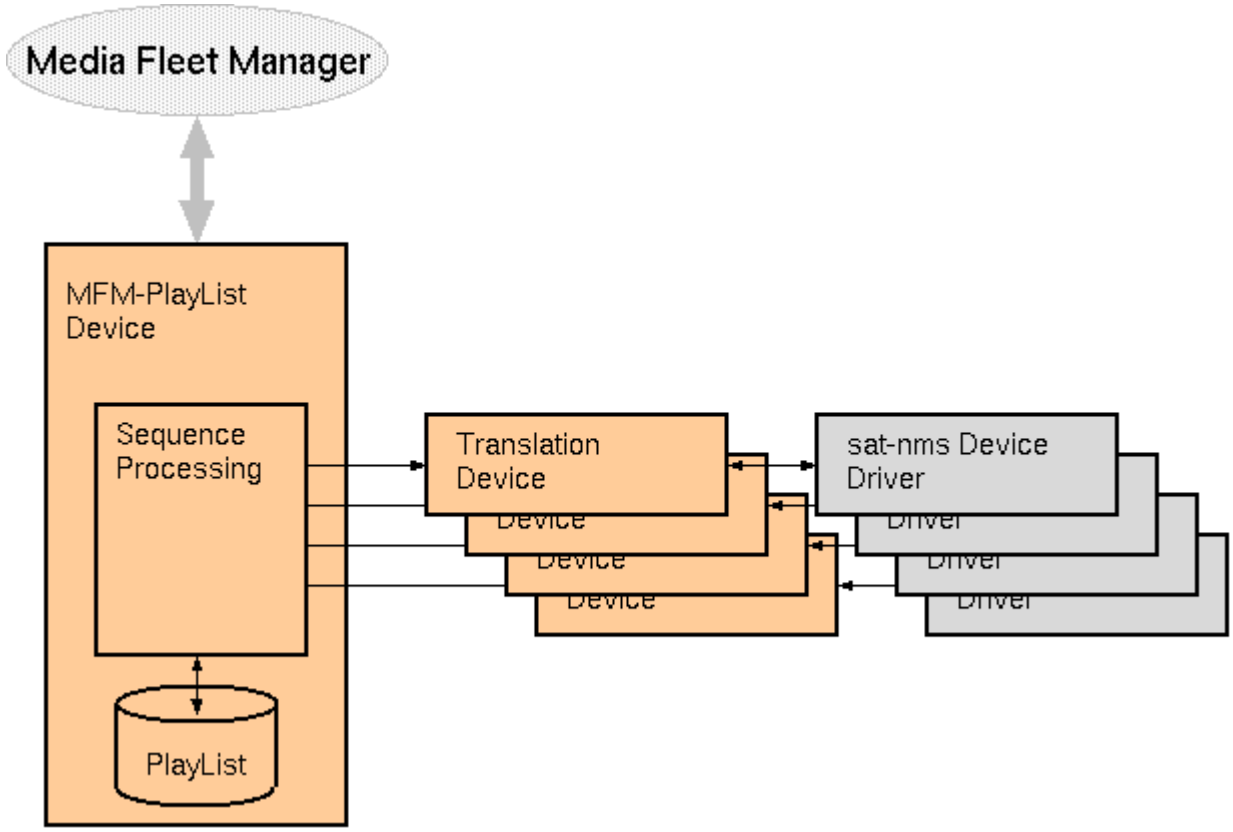
9.338 MFM-SkyWANIdu

The MFM-SkyWANIdu logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager. The MFM-SkyWANIdu is a so called translation device which interconnects the MFM-PlayList device to a sat-nms SkyWAN IDU device driver. The MFM-SkyWANIdu logical device must be configured together with a NDSatCom-SkyWAN-IDU device driver.

Internally the Media Fleet Manager codes all equipment settings in binary enumerations (for choice parameters) and scaled integers for numeric values. The MFM-SkyWANIdu translation device takes these settings and translates them into the parameter representation expected by the sat-nms device driver. For this, it inspects each parameter's range definition and matches the settings against the MFM encoding of the parameters.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General status.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
aliasName	The alias name of the device reported to the Media Fleet Manager.
checkDelay	After receiving a set of parameters from the MFM-Playlist and setting these parameters at the IRD, the MFM-SkyWANIdu device waits the time set here until it checks if all parameters have been set properly.
onAirFaultDelay	After going on air, the device suppresses faults for the time configured here. This helps to avoid receiver lock faults/warnings for the first seconds after the signal is switched on.
skywanDevice	The device name of the SkyWAN IDU. This is used to get the IRD's summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.

Remarks

- The device does not provide individual alarm flags. It reports a summary fault state which resembles the state of the device configured with the 'skywanDevice' configuration setting.
- Setting the device to the FAULT-SUPPRESSED mode will suppress alarms or warnings of the IRD which would normally lead to ONAIR-ALARM or ONAIR-WARNING alerts at the Media Fleet Manager. FAULT-SUPPRESSED does not suppress the communication fault check the MFM-PlayList device does before configuring the device nor does it suppress the check of correct parameter setting at the end of the configuration process.
- Setting the device OUT-OF-SERVICE switches it to a simulation mode in which it does not send any command to the IRD and reports to the MFM-PlayList 'everything OK' all the time.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.aliasName	TEXT	SAVE SETUP	StringRange
config.checkDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.onAirFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.skywanDevice	TEXT	SAVE SETUP	StringRange
set	TEXT		StringRange
check	CHOICE	R/O	EnumRange R/O (IDLE PENDING SUCCESS FAILED)
errorCode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
onAir	CHOICE		EnumRange (OFFAIR ONAIR)
releaseCheckFault	TEXT		StringRange

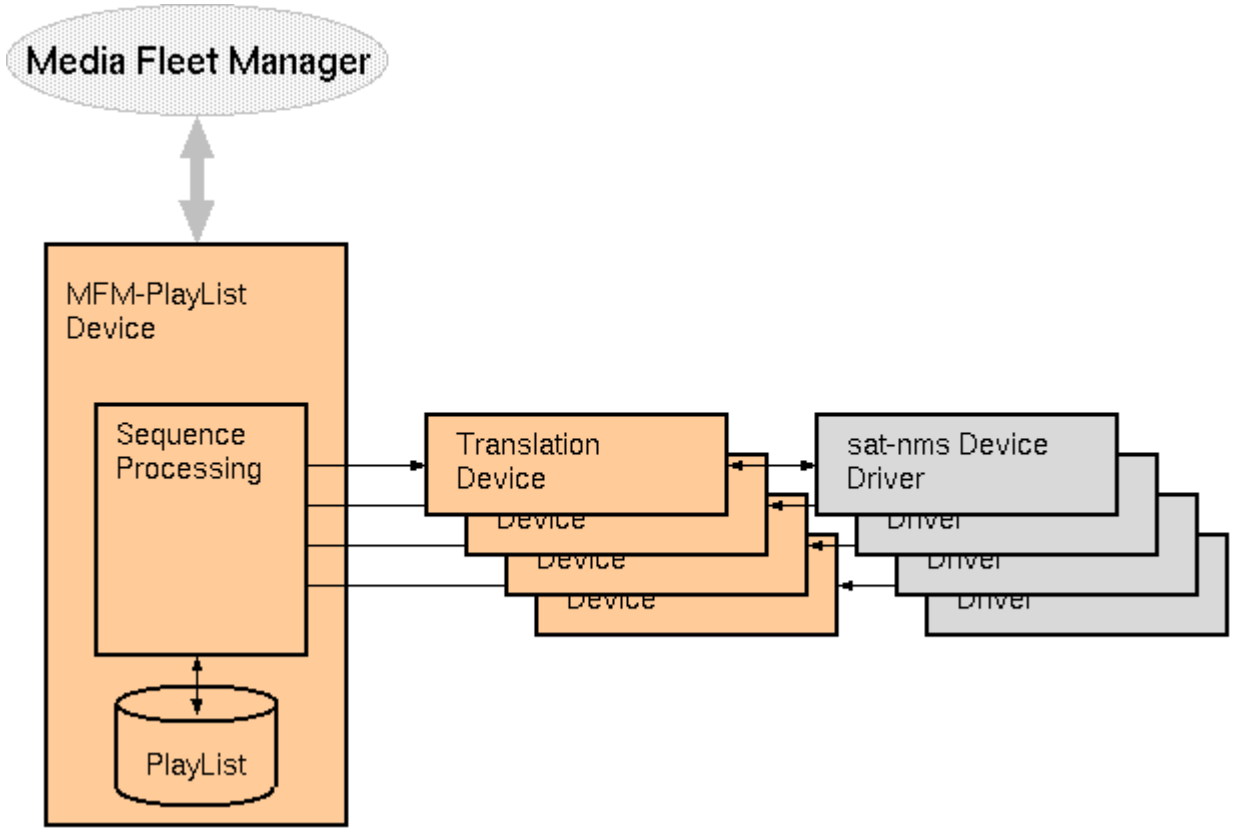
9.339 MFM-Upc

The MFM-Upc logical device is part of the sat-nms MFM framework which interconnects a sat-nms VLC to the ND SatCom Media Fleet Manager.The MFM-Upc is a so called translation device which interconnects the MFM-PlayList device to the sat-nms device drivers of an encoder / modulator / upconverter / amplifier uplink chain. As the message ID of each target parameter is individually configurable, the device seamlessly supports combined devices like encoders with built in modulators.

Internally the Media Fleet Manager codes all equipment settings in binary enumerations (for choice parameters) and scaled integers for numeric values. The MFM-Upc translation device takes these settings and translates them into the parameter representation expected by the sat-nms device drivers. For this, it inspects each parameter's range definition and matches the settings against the MFM encoding of the parameters.

Overview

The sat-nms MFM framework consists of the MFM-PlayList device and a number of so called translation devices. The MFM-PlayList device does the communication with the Media Fleet Manager using the proprietary protocol of this. It manages the sequence control of transmissions, even if there is no connection to the MFM.



Beside this, there are translation devices which have to be configured with one instance for each DVB uplink or downlink chain, for each IP encoder or decoder. The task of translation devices is to translate the settings commanded by the media fleet manager in it's own format into the individual format each sat-nms device driver expects. The translation devices also check if parameters have been set properly and forward this information to the MFM-PlayList device.

Adding a translation device for a certain physical device to the configuration automatically makes this device known to the MFM. The following types of translation devices are available with the sat-nms MFM framework:

- MFM-IpDecoder --- Connects an IP decoder device to the MFM-PlayList.
- MFM-IpEncoder --- Connects an IP encoder device to the MFM-PlayList.
- MFM-Ird --- Connects a (DVB) IRD to the MFM-PlayList.
- MFM-PlayList --- Connects a satellite modem for intercom purposes to the MFM-PlayList.
- MFM-SkyWANIdu --- Connects a ND SatCom SkyWAN IDU to the MFM-PlayList.
- MFM-Upc --- Connects a (DVB) uplink chain consisting of encoder, modulator, upconverter and power amplifier to the MFM-PlayList.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- General status.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the page 'Maintenance' the setup dialog contains some settings to configure this logical device.

parameter	***description***
aliasName	The alias name of the device reported to the Media Fleet Manager
checkDelay	After receiving a set of parameters from the MFM-Playlist and setting these parameters at the IRD, the MFM-Upc device waits the time set here until it checks if all parameters have been set properly.
onAirFaultDelay	After going on air, the device suppresses faults for the time configured here. This helps to avoid receiver lock faults/warnings for the first seconds after the signal is switched on.
upcDevice	The device name of the transmit chain's upconverter. This is used to get the device summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
encDevice	The device name of the transmit chain's encoder. This is used to get the device summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
hpaDevice	The device name of the transmit chain's HPA. This is used to get the device summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
modDevice	The device name of the transmit chain's modulator. This is used to get the device summary fault, it's communications fault and it's OPERATING/OUT-OF-SERVICE setting.
frequencyId	The message ID of the parameter controlling the transmit frequency. The frequency is commanded in terms of true satellite frequency, so either the target device or it's device driver has to do the LO calculations.
polarizationId	The message ID of the parameter controlling the transmit polarization plane.

parameter	***description***
eirpId	The message ID of the parameter controlling the transmit power. The power is commanded in terms EIRP (dBW), so the target device for this parameter usually will be a power calculating / controlling logical device like the Level-Set device.
measuredEirpId	The message ID of a parameter reporting the measured transmit power. The power is expected in terms EIRP (dBW). The Level-Set provides a measurement value which can be used for this.
symbolRateId	The message ID of the modulator parameter controlling the symbol rate.
dvbModeId	The message ID of the modulator parameter controlling the dvb mode (DVB-S/DVBS2).
modulationId	The message ID of the modulator parameter controlling the modulation type (QPSK, 8PSK, etc).
fecId	The message ID of the modulator parameter controlling the FEC.
pilotsId	The message ID of the modulator parameter switching DVB-S2 pilots on/off.
rollOffId	The message ID of the modulator parameter controlling the roll off factor.
frameSizeId	The message ID of the modulator parameter controlling the DVB frame size (NORMAL/SHORT)
packetSizeId	The message ID of the modulator parameter controlling the packet size (188/204)
encryptModeId	The message ID of the encoder parameter controlling the encryption mode (BISS-1/BISS-E).
session1WordId	The message ID of the encoder parameter setting the BISS-1 session word.
sessionEWordId	The message ID of the encoder parameter setting the BISS-E session word.
injectedIDId	The message ID of the encoder parameter setting the BISS injected ID.
rasKeyId	The message ID of the encoder parameter setting the RAS encryption key.
audioRateId	The message ID of the encoder parameter setting the first audio channel bitrate.

parameter	*description*
audioRate2Id	The message ID of the encoder parameter setting the second audio channel bitrate.
audio2EnableId	The message ID of the encoder parameter enabling the second audio channel.
videoRateId	The message ID of the encoder parameter setting the video bitrate.
rfOnId	The message ID of the parameter switching transmit power ON/OFF.

All message ID settings above may be left empty. The MFM-Upc device will not control a parameter with an empty ID setting and it will not check if it is set properly.

If the transmit chain uses a combined encoder / modulator, the 'modDevice' setting may be left empty. The same applies to the 'upcDevice' and 'hpaDevice' settings if they are not present as separately controlled devices.

Session word and encryption key parameters are not read back and compared to the commanded values, as some devices do not support a readout of these values.

Video profiles

Encoder devices are handling encoding profile parameters in different ways, depending on vendor and / or encoder type. For this reason there is no direct translation possible from the MFM specific parameters selecting an encoding profile and the corresponding encoder parameters. Instead, the MFM-Upc device calls a sat-nms device preset to configure these parameters.

To make this work, you have to store device presets for each particular encoder type you are using and the video encoding profiles used in your MFM equipment presets. The names of the presets called by the MFM-Upc device follow a fixed scheme, resembling the combination of the "video coding standard" and "coding format" MFM parameters.

This is the list of all presets names recognized by the MFM-Upc device:

```
MPEG2-MP@ML-420     MPEG4-MP@ML-420
MPEG2-MP@LL-420     MPEG4-MP@LL-420
MPEG2-HP@ML-422     MPEG4-HP@ML-422
MPEG2-HP@LL-422     MPEG4-HP@LL-422
MPEG2-SP@LL-420     MPEG4-SP@LL-420
MPEG2-SP@ML-420     MPEG4-SP@ML-420
MPEG2-MP@H14-420    MPEG4-MP@H14-420
MPEG2-MP@HL-420     MPEG4-MP@HL-420
MPEG2-HP@H14-422    MPEG4-HP@H14-422
MPEG2-HP@HL-422     MPEG4-HP@HL-422
MPEG2-HP@HL-420     MPEG4-HP@HL-420
MPEG2-HP@H14-420    MPEG4-HP@H14-420
MPEG2-HP@ML-420     MPEG4-HP@ML-420
```

It is not necessary to create all of these presets, only these ones addressed by MFM transmission presets are required.

If the presets are created by the "preset store" function in the device window, they should be edited afterwards to remove all parameters from the preset which are set explicitly by the MFM-Upc device. If such a parameter remains in the preset, it may overwrite the value actually commanded by the MFM-Upc device. There is no fixed order in which preset contained parameters and explicitly commanded ones are processed by the encoder device driver.

Remarks

- The device does not provide individual alarm flags. It reports a summary fault state which resembles the summary state of the devices configured with the 'encDevice', 'modDevice', 'upcDevice' and 'hpaDevice' configuration settings.
- Setting the device to the FAULT-SUPPRESSED mode will suppress alarms or warnings of the attached devices which would normally lead to ONAIR-ALARM or ONAIR-WARNING alerts at the Media Fleet Manager. FAULT-SUPPRESSED does not suppress the communication fault checks the MFM-PlayList device does before configuring the device nor does it suppress the check of correct parameter setting at the end of the configuration process.
- Setting the device OUT-OF-SERVICE switches it to a simulation mode in which it does not send commands to any device and reports to the MFM-PlayList 'everything OK' all the time.
- If the modulator does not provide separate parameters for DVB mode, modulation type end FEC but uses a combined MODCOD Setting for this, you may want to use the Modcod-Splitter device to control these parameters.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.aliasName	TEXT	SAVE SETUP	StringRange
config.checkDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.onAirFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.upcDevice	TEXT	SAVE SETUP	StringRange
config.encDevice	TEXT	SAVE SETUP	StringRange
config.hpaDevice	TEXT	SAVE SETUP	StringRange
config.modDevice	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.measuredEirpId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.fecId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.frameSizeId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange
config.encryptModeId	TEXT	SAVE SETUP	StringRange
config.session1WordId	TEXT	SAVE SETUP	StringRange
config.sessionEWordId	TEXT	SAVE SETUP	StringRange
config.injectedIDId	TEXT	SAVE SETUP	StringRange
config.rasKeyId	TEXT	SAVE SETUP	StringRange
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audioRate2Id	TEXT	SAVE SETUP	StringRange
config.audio2EnableId	TEXT	SAVE SETUP	StringRange
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.rfOnId	TEXT	SAVE SETUP	StringRange
set	TEXT		StringRange
check	CHOICE	R/O	EnumRange R/O (IDLE PENDING SUCCESS FAILED)

name	*type*	*flags*	*range*
errorCode	INTEGER	R/O	IntegerRange R/O (0 .. 0)
onAir	CHOICE		EnumRange (OFFAIR ONAIR)
releaseCheckFault	TEXT		StringRange
rfOn	BOOLEAN	R/O	BooleanRange R/O (true false)
measuredEirp	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.340 MIB2-Ethernet-Switch

The *MIB2-Ethernet-Switch* device driver monitors the state of up to 48 interfaces of a SNMP device which implements the mib-2.interfaces table. Compared to the MIB2-Interface-Monitor it monitors a variety of interface statistics e.g. operational and admin state, link speed, in/out bytes, in/out errors, rx and tx rate etc.

The driver permits to configure the list of interfaces to be monitored. For this, the table index of each interface to monitor must be entered at the maintenance/setup page of the device window. As a side effect, devices which are numbering their interfaces non-sequentially are supported as well.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Overview page showing interface state and rx and tx rate for up to 48 ports
- --- Detailed interface statistics for Interface 1-8
- --- Detailed interface statistics for Interface 9-16
- --- Detailed interface statistics for Interface 17-24
- --- Detailed interface statistics for Interface 25-32
- --- Detailed interface statistics for Interface 33-40
- --- Detailed interface statistics for Interface 41-48
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the redundancy switch in the sat-nms software.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ifIndex.1 ... ifIndex.48 --- For each interface to be monitored, the table index of the interface in the mib-2.interfaces has to be entered here. You find the index values with the documentation of the router or switch to monitor or simply by "walking" through the interfaces list of the device by means of a SNMP MIB browser tool. Setting an index value to zero means not to monitor this interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.ifIndex.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.7	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.8	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.9	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.17	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.18	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.19	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.20	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.21	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.22	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.23	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.24	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.25	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.26	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.27	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.28	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.29	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.30	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.31	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.32	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.33	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.34	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.35	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.36	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.37	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.38	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.39	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.40	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.41	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.42	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.43	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.44	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.45	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.46	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.47	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.48	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
info.sysDescription	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysContact	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
if.load	CHOICE	R/O	EnumRange R/O (UNDEF OFF OK WARNING ALARM)
internal.l	CHOICE	R/O	EnumRange R/O (UNDEF OFF OK WARNING ALARM)
if.01.comment	TEXT	SAVE	StringRange
if.01.descr	TEXT	R/O	StringRange R/O DIS
if.01.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.physAddress	TEXT	R/O	StringRange R/O DIS
if.01.adminStatus	TEXT	R/O	StringRange R/O DIS
if.01.operStatus	TEXT	R/O	StringRange R/O DIS
if.01.status	TEXT	R/O	StringRange R/O DIS
if.01.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.01.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.01.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.01.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.01.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.02.comment	TEXT	SAVE	StringRange
if.02.descr	TEXT	R/O	StringRange R/O DIS
if.02.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.02.physAddress	TEXT	R/O	StringRange R/O DIS
if.02.adminStatus	TEXT	R/O	StringRange R/O DIS
if.02.status	TEXT	R/O	StringRange R/O DIS
if.02.operStatus	TEXT	R/O	StringRange R/O DIS
if.02.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.02.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.02.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.02.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.02.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.03.comment	TEXT	SAVE	StringRange
if.03.descr	TEXT	R/O	StringRange R/O DIS
if.03.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.physAddress	TEXT	R/O	StringRange R/O DIS
if.03.adminStatus	TEXT	R/O	StringRange R/O DIS
if.03.operStatus	TEXT	R/O	StringRange R/O DIS
if.03.status	TEXT	R/O	StringRange R/O DIS
if.03.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.03.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.03.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.03.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.03.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.03.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.04.comment	TEXT	SAVE	StringRange
if.04.descr	TEXT	R/O	StringRange R/O DIS
if.04.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.physAddress	TEXT	R/O	StringRange R/O DIS
if.04.adminStatus	TEXT	R/O	StringRange R/O DIS
if.04.operStatus	TEXT	R/O	StringRange R/O DIS
if.04.status	TEXT	R/O	StringRange R/O DIS
if.04.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.04.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.04.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.04.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.04.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.04.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.05.comment	TEXT	SAVE	StringRange
if.05.descr	TEXT	R/O	StringRange R/O DIS
if.05.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.physAddress	TEXT	R/O	StringRange R/O DIS
if.05.adminStatus	TEXT	R/O	StringRange R/O DIS
if.05.operStatus	TEXT	R/O	StringRange R/O DIS
if.05.status	TEXT	R/O	StringRange R/O DIS
if.05.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.05.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.05.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.05.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.05.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.06.comment	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
if.06.descr	TEXT	R/O	StringRange R/O DIS
if.06.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.physAddress	TEXT	R/O	StringRange R/O DIS
if.06.adminStatus	TEXT	R/O	StringRange R/O DIS
if.06.operStatus	TEXT	R/O	StringRange R/O DIS
if.06.status	TEXT	R/O	StringRange R/O DIS
if.06.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.06.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.06.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.06.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.06.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.07.comment	TEXT	SAVE	StringRange
if.07.descr	TEXT	R/O	StringRange R/O DIS
if.07.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.physAddress	TEXT	R/O	StringRange R/O DIS
if.07.adminStatus	TEXT	R/O	StringRange R/O DIS
if.07.operStatus	TEXT	R/O	StringRange R/O DIS
if.07.status	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.07.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.07.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.07.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.07.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.07.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.08.comment	TEXT	SAVE	StringRange
if.08.descr	TEXT	R/O	StringRange R/O DIS
if.08.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.physAddress	TEXT	R/O	StringRange R/O DIS
if.08.adminStatus	TEXT	R/O	StringRange R/O DIS
if.08.operStatus	TEXT	R/O	StringRange R/O DIS
if.08.status	TEXT	R/O	StringRange R/O DIS
if.08.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.08.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.08.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.08.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.08.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.08.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.09.comment	TEXT	SAVE	StringRange
if.09.descr	TEXT	R/O	StringRange R/O DIS
if.09.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.physAddress	TEXT	R/O	StringRange R/O DIS
if.09.adminStatus	TEXT	R/O	StringRange R/O DIS
if.09.operStatus	TEXT	R/O	StringRange R/O DIS
if.09.status	TEXT	R/O	StringRange R/O DIS
if.09.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.09.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.09.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.09.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)

name	*type*	*flags*	*range*
if.09.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.10.comment	TEXT	SAVE	StringRange
if.10.descr	TEXT	R/O	StringRange R/O DIS
if.10.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.physAddress	TEXT	R/O	StringRange R/O DIS
if.10.adminStatus	TEXT	R/O	StringRange R/O DIS
if.10.status	TEXT	R/O	StringRange R/O DIS
if.10.operStatus	TEXT	R/O	StringRange R/O DIS
if.10.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.10.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.10.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.10.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.10.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.11.comment	TEXT	SAVE	StringRange
if.11.descr	TEXT	R/O	StringRange R/O DIS
if.11.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.physAddress	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.11.adminStatus	TEXT	R/O	StringRange R/O DIS
if.11.operStatus	TEXT	R/O	StringRange R/O DIS
if.11.status	TEXT	R/O	StringRange R/O DIS
if.11.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.11.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.11.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.11.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.11.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.12.comment	TEXT	SAVE	StringRange
if.12.descr	TEXT	R/O	StringRange R/O DIS
if.12.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.physAddress	TEXT	R/O	StringRange R/O DIS
if.12.adminStatus	TEXT	R/O	StringRange R/O DIS
if.12.operStatus	TEXT	R/O	StringRange R/O DIS
if.12.status	TEXT	R/O	StringRange R/O DIS
if.12.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.12.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.12.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.12.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.12.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.12.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.13.comment	TEXT	SAVE	StringRange
if.13.descr	TEXT	R/O	StringRange R/O DIS
if.13.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.physAddress	TEXT	R/O	StringRange R/O DIS
if.13.adminStatus	TEXT	R/O	StringRange R/O DIS
if.13.operStatus	TEXT	R/O	StringRange R/O DIS
if.13.status	TEXT	R/O	StringRange R/O DIS
if.13.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.13.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
if.13.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.13.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.13.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.14.comment	TEXT	SAVE	StringRange
if.14.descr	TEXT	R/O	StringRange R/O DIS
if.14.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.physAddress	TEXT	R/O	StringRange R/O DIS
if.14.adminStatus	TEXT	R/O	StringRange R/O DIS
if.14.operStatus	TEXT	R/O	StringRange R/O DIS
if.14.status	TEXT	R/O	StringRange R/O DIS
if.14.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.14.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.14.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.14.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.14.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.15.comment	TEXT	SAVE	StringRange
if.15.descr	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.15.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.physAddress	TEXT	R/O	StringRange R/O DIS
if.15.adminStatus	TEXT	R/O	StringRange R/O DIS
if.15.operStatus	TEXT	R/O	StringRange R/O DIS
if.15.status	TEXT	R/O	StringRange R/O DIS
if.15.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.15.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.15.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.15.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.15.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.16.comment	TEXT	SAVE	StringRange
if.16.descr	TEXT	R/O	StringRange R/O DIS
if.16.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.physAddress	TEXT	R/O	StringRange R/O DIS
if.16.adminStatus	TEXT	R/O	StringRange R/O DIS
if.16.operStatus	TEXT	R/O	StringRange R/O DIS
if.16.status	TEXT	R/O	StringRange R/O DIS
if.16.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.16.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.16.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.16.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.16.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.16.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.17.comment	TEXT	SAVE	StringRange
if.17.descr	TEXT	R/O	StringRange R/O DIS
if.17.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.physAddress	TEXT	R/O	StringRange R/O DIS
if.17.adminStatus	TEXT	R/O	StringRange R/O DIS
if.17.operStatus	TEXT	R/O	StringRange R/O DIS
if.17.status	TEXT	R/O	StringRange R/O DIS
if.17.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.17.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.17.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.17.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.17.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.17.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.18.comment	TEXT	SAVE	StringRange
if.18.descr	TEXT	R/O	StringRange R/O DIS
if.18.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.physAddress	TEXT	R/O	StringRange R/O DIS
if.18.adminStatus	TEXT	R/O	StringRange R/O DIS
if.18.status	TEXT	R/O	StringRange R/O DIS
if.18.operStatus	TEXT	R/O	StringRange R/O DIS
if.18.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.18.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.18.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.18.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.18.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)

name	*type*	*flags*	*range*
if.19.comment	TEXT	SAVE	StringRange
if.19.descr	TEXT	R/O	StringRange R/O DIS
if.19.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.physAddress	TEXT	R/O	StringRange R/O DIS
if.19.adminStatus	TEXT	R/O	StringRange R/O DIS
if.19.operStatus	TEXT	R/O	StringRange R/O DIS
if.19.status	TEXT	R/O	StringRange R/O DIS
if.19.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.19.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.19.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.19.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.19.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.20.comment	TEXT	SAVE	StringRange
if.20.descr	TEXT	R/O	StringRange R/O DIS
if.20.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.physAddress	TEXT	R/O	StringRange R/O DIS
if.20.adminStatus	TEXT	R/O	StringRange R/O DIS
if.20.operStatus	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.20.status	TEXT	R/O	StringRange R/O DIS
if.20.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.20.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.20.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.20.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.20.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.21.comment	TEXT	SAVE	StringRange
if.21.descr	TEXT	R/O	StringRange R/O DIS
if.21.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.physAddress	TEXT	R/O	StringRange R/O DIS
if.21.adminStatus	TEXT	R/O	StringRange R/O DIS
if.21.operStatus	TEXT	R/O	StringRange R/O DIS
if.21.status	TEXT	R/O	StringRange R/O DIS
if.21.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.21.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.21.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.21.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.21.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.21.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.22.comment	TEXT	SAVE	StringRange
if.22.descr	TEXT	R/O	StringRange R/O DIS
if.22.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.physAddress	TEXT	R/O	StringRange R/O DIS
if.22.adminStatus	TEXT	R/O	StringRange R/O DIS
if.22.operStatus	TEXT	R/O	StringRange R/O DIS
if.22.status	TEXT	R/O	StringRange R/O DIS
if.22.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.22.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.22.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
if.22.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.22.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.23.comment	TEXT	SAVE	StringRange
if.23.descr	TEXT	R/O	StringRange R/O DIS
if.23.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.physAddress	TEXT	R/O	StringRange R/O DIS
if.23.adminStatus	TEXT	R/O	StringRange R/O DIS
if.23.operStatus	TEXT	R/O	StringRange R/O DIS
if.23.status	TEXT	R/O	StringRange R/O DIS
if.23.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.23.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.23.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.23.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.23.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.24.comment	TEXT	SAVE	StringRange
if.24.descr	TEXT	R/O	StringRange R/O DIS
if.24.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.24.physAddress	TEXT	R/O	StringRange R/O DIS
if.24.adminStatus	TEXT	R/O	StringRange R/O DIS
if.24.operStatus	TEXT	R/O	StringRange R/O DIS
if.24.status	TEXT	R/O	StringRange R/O DIS
if.24.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.24.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.24.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.24.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.24.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.25.comment	TEXT	SAVE	StringRange
if.25.descr	TEXT	R/O	StringRange R/O DIS
if.25.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.physAddress	TEXT	R/O	StringRange R/O DIS
if.25.adminStatus	TEXT	R/O	StringRange R/O DIS
if.25.operStatus	TEXT	R/O	StringRange R/O DIS
if.25.status	TEXT	R/O	StringRange R/O DIS
if.25.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.25.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.25.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.25.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.25.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.25.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.26.comment	TEXT	SAVE	StringRange
if.26.descr	TEXT	R/O	StringRange R/O DIS
if.26.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.physAddress	TEXT	R/O	StringRange R/O DIS
if.26.adminStatus	TEXT	R/O	StringRange R/O DIS
if.26.status	TEXT	R/O	StringRange R/O DIS
if.26.operStatus	TEXT	R/O	StringRange R/O DIS
if.26.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.26.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.26.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.26.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.26.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.26.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.27.comment	TEXT	SAVE	StringRange
if.27.descr	TEXT	R/O	StringRange R/O DIS
if.27.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.physAddress	TEXT	R/O	StringRange R/O DIS
if.27.adminStatus	TEXT	R/O	StringRange R/O DIS
if.27.operStatus	TEXT	R/O	StringRange R/O DIS
if.27.status	TEXT	R/O	StringRange R/O DIS
if.27.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.27.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.27.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.27.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.27.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.28.comment	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
if.28.descr	TEXT	R/O	StringRange R/O DIS
if.28.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.physAddress	TEXT	R/O	StringRange R/O DIS
if.28.adminStatus	TEXT	R/O	StringRange R/O DIS
if.28.operStatus	TEXT	R/O	StringRange R/O DIS
if.28.status	TEXT	R/O	StringRange R/O DIS
if.28.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.28.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.28.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.28.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.28.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.29.comment	TEXT	SAVE	StringRange
if.29.descr	TEXT	R/O	StringRange R/O DIS
if.29.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.physAddress	TEXT	R/O	StringRange R/O DIS
if.29.adminStatus	TEXT	R/O	StringRange R/O DIS
if.29.operStatus	TEXT	R/O	StringRange R/O DIS
if.29.status	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.29.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.29.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.29.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.29.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.29.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.30.comment	TEXT	SAVE	StringRange
if.30.descr	TEXT	R/O	StringRange R/O DIS
if.30.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.physAddress	TEXT	R/O	StringRange R/O DIS
if.30.adminStatus	TEXT	R/O	StringRange R/O DIS
if.30.operStatus	TEXT	R/O	StringRange R/O DIS
if.30.status	TEXT	R/O	StringRange R/O DIS
if.30.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.30.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.30.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.30.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.30.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.30.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.31.comment	TEXT	SAVE	StringRange
if.31.descr	TEXT	R/O	StringRange R/O DIS
if.31.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.physAddress	TEXT	R/O	StringRange R/O DIS
if.31.adminStatus	TEXT	R/O	StringRange R/O DIS
if.31.operStatus	TEXT	R/O	StringRange R/O DIS
if.31.status	TEXT	R/O	StringRange R/O DIS
if.31.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.31.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.31.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.31.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)

name	*type*	*flags*	*range*
if.31.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.32.comment	TEXT	SAVE	StringRange
if.32.descr	TEXT	R/O	StringRange R/O DIS
if.32.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.physAddress	TEXT	R/O	StringRange R/O DIS
if.32.adminStatus	TEXT	R/O	StringRange R/O DIS
if.32.operStatus	TEXT	R/O	StringRange R/O DIS
if.32.status	TEXT	R/O	StringRange R/O DIS
if.32.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.32.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.32.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.32.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.32.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.33.comment	TEXT	SAVE	StringRange
if.33.descr	TEXT	R/O	StringRange R/O DIS
if.33.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.physAddress	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.33.adminStatus	TEXT	R/O	StringRange R/O DIS
if.33.operStatus	TEXT	R/O	StringRange R/O DIS
if.33.status	TEXT	R/O	StringRange R/O DIS
if.33.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.33.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.33.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.33.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.33.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.34.comment	TEXT	SAVE	StringRange
if.34.descr	TEXT	R/O	StringRange R/O DIS
if.34.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.physAddress	TEXT	R/O	StringRange R/O DIS
if.34.adminStatus	TEXT	R/O	StringRange R/O DIS
if.34.operStatus	TEXT	R/O	StringRange R/O DIS
if.34.status	TEXT	R/O	StringRange R/O DIS
if.34.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.34.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.34.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.34.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.34.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.34.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.35.comment	TEXT	SAVE	StringRange
if.35.descr	TEXT	R/O	StringRange R/O DIS
if.35.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.physAddress	TEXT	R/O	StringRange R/O DIS
if.35.adminStatus	TEXT	R/O	StringRange R/O DIS
if.35.operStatus	TEXT	R/O	StringRange R/O DIS
if.35.status	TEXT	R/O	StringRange R/O DIS
if.35.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.35.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
if.35.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.35.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.35.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.36.comment	TEXT	SAVE	StringRange
if.36.descr	TEXT	R/O	StringRange R/O DIS
if.36.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.physAddress	TEXT	R/O	StringRange R/O DIS
if.36.adminStatus	TEXT	R/O	StringRange R/O DIS
if.36.operStatus	TEXT	R/O	StringRange R/O DIS
if.36.status	TEXT	R/O	StringRange R/O DIS
if.36.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.36.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.36.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.36.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.36.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.37.comment	TEXT	SAVE	StringRange
if.37.descr	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.37.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.physAddress	TEXT	R/O	StringRange R/O DIS
if.37.adminStatus	TEXT	R/O	StringRange R/O DIS
if.37.operStatus	TEXT	R/O	StringRange R/O DIS
if.37.status	TEXT	R/O	StringRange R/O DIS
if.37.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.37.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.37.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.37.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.37.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.38.comment	TEXT	SAVE	StringRange
if.38.descr	TEXT	R/O	StringRange R/O DIS
if.38.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.physAddress	TEXT	R/O	StringRange R/O DIS
if.38.adminStatus	TEXT	R/O	StringRange R/O DIS
if.38.status	TEXT	R/O	StringRange R/O DIS
if.38.operStatus	TEXT	R/O	StringRange R/O DIS
if.38.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.38.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.38.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.38.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.38.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.38.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.39.comment	TEXT	SAVE	StringRange
if.39.descr	TEXT	R/O	StringRange R/O DIS
if.39.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.physAddress	TEXT	R/O	StringRange R/O DIS
if.39.adminStatus	TEXT	R/O	StringRange R/O DIS
if.39.operStatus	TEXT	R/O	StringRange R/O DIS
if.39.status	TEXT	R/O	StringRange R/O DIS
if.39.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.39.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.39.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.39.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.39.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.39.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.40.comment	TEXT	SAVE	StringRange
if.40.descr	TEXT	R/O	StringRange R/O DIS
if.40.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.physAddress	TEXT	R/O	StringRange R/O DIS
if.40.adminStatus	TEXT	R/O	StringRange R/O DIS
if.40.operStatus	TEXT	R/O	StringRange R/O DIS
if.40.status	TEXT	R/O	StringRange R/O DIS
if.40.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.40.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.40.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.40.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)

name	*type*	*flags*	*range*
if.40.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.41.comment	TEXT	SAVE	StringRange
if.41.descr	TEXT	R/O	StringRange R/O DIS
if.41.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.physAddress	TEXT	R/O	StringRange R/O DIS
if.41.adminStatus	TEXT	R/O	StringRange R/O DIS
if.41.operStatus	TEXT	R/O	StringRange R/O DIS
if.41.status	TEXT	R/O	StringRange R/O DIS
if.41.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.41.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.41.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.41.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.41.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.42.comment	TEXT	SAVE	StringRange
if.42.descr	TEXT	R/O	StringRange R/O DIS
if.42.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.physAddress	TEXT	R/O	StringRange R/O DIS
if.42.adminStatus	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
if.42.operStatus	TEXT	R/O	StringRange R/O DIS
if.42.status	TEXT	R/O	StringRange R/O DIS
if.42.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.42.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.42.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.42.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.42.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.43.comment	TEXT	SAVE	StringRange
if.43.descr	TEXT	R/O	StringRange R/O DIS
if.43.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.physAddress	TEXT	R/O	StringRange R/O DIS
if.43.adminStatus	TEXT	R/O	StringRange R/O DIS
if.43.operStatus	TEXT	R/O	StringRange R/O DIS
if.43.status	TEXT	R/O	StringRange R/O DIS
if.43.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.43.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.43.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.43.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.43.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.43.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.44.comment	TEXT	SAVE	StringRange
if.44.descr	TEXT	R/O	StringRange R/O DIS
if.44.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.physAddress	TEXT	R/O	StringRange R/O DIS
if.44.adminStatus	TEXT	R/O	StringRange R/O DIS
if.44.operStatus	TEXT	R/O	StringRange R/O DIS
if.44.status	TEXT	R/O	StringRange R/O DIS
if.44.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.44.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.44.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
if.44.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.44.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.45.comment	TEXT	SAVE	StringRange
if.45.descr	TEXT	R/O	StringRange R/O DIS
if.45.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.physAddress	TEXT	R/O	StringRange R/O DIS
if.45.adminStatus	TEXT	R/O	StringRange R/O DIS
if.45.operStatus	TEXT	R/O	StringRange R/O DIS
if.45.status	TEXT	R/O	StringRange R/O DIS
if.45.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.45.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.45.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.45.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.45.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.46.comment	TEXT	SAVE	StringRange
if.46.descr	TEXT	R/O	StringRange R/O DIS
if.46.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.46.physAddress	TEXT	R/O	StringRange R/O DIS
if.46.adminStatus	TEXT	R/O	StringRange R/O DIS
if.46.status	TEXT	R/O	StringRange R/O DIS
if.46.operStatus	TEXT	R/O	StringRange R/O DIS
if.46.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.46.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.46.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.46.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.46.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.47.comment	TEXT	SAVE	StringRange
if.47.descr	TEXT	R/O	StringRange R/O DIS
if.47.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.physAddress	TEXT	R/O	StringRange R/O DIS
if.47.adminStatus	TEXT	R/O	StringRange R/O DIS
if.47.operStatus	TEXT	R/O	StringRange R/O DIS
if.47.status	TEXT	R/O	StringRange R/O DIS
if.47.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.47.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.47.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.47.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.47.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.47.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.48.comment	TEXT	SAVE	StringRange
if.48.descr	TEXT	R/O	StringRange R/O DIS
if.48.speed	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.physAddress	TEXT	R/O	StringRange R/O DIS
if.48.adminStatus	TEXT	R/O	StringRange R/O DIS
if.48.operStatus	TEXT	R/O	StringRange R/O DIS
if.48.status	TEXT	R/O	StringRange R/O DIS
if.48.inBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.outBytes	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.inErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.outErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.inDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.outDropped	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.inUnknown	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
if.48.outQueue	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
if.48.rxRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.48.txRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
if.48.rxLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
if.48.txLoad	CHOICE	R/O	EnumRange R/O DIS (UNDEF OFF OK WARNING ALARM)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.o	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.341 MIB2-Interface-Monitor

The *MIB2-Interface-Monitor* device driver monitors the state of up to 48 interfaces of a SNMP device which implements the mib-2.interfaces table. In addition to a textual description of the operational state (ifOperStatus) of each interface monitored, the driver provides fault signals for each interface not being 'up'. When you require more details on each interface please try using the following alternative driver MIB2-Ethernet-Switch.

The driver permits to configure the list of interfaces to be monitored. For this, the table index of each interface to monitor must be entered at the maintenance/setup page of the device window. As a side effect, devices which are numbering their interfaces non-sequentially are supported as well.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational state of each interface monitored.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the redundancy switch in the sat-nms software.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ifIndex.1 ... ifIndex.8 --- For each interface to be monitored, the table index of the interface in the mib-2.interfaces has to be entered here. You find the index values with the documentation of the router or switch to monitor or simply by "walking" through the interfaces list of the device by means of a SNMP MIB browser tool. Setting an index value to zero means not to monitor this value.

Fault handling

For each of the 8 monitorable interfaces, the device provides a fault flag which is set if this interface is not "up". You may suppress this behaviour by setting the appropriate fault flags to "OFF" or "INFO", depending on the type of signalling desired for each individual interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.sysDescr	TEXT	R/O	StringRange R/O
config.ifIndex.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.7	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.8	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.9	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.17	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.18	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.19	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.20	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.21	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.22	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.23	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.24	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.25	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.26	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.27	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.28	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.29	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.30	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.31	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.32	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.33	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.34	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.35	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.36	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.37	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.38	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.39	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.40	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.41	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.42	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.43	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.44	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.45	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.46	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.47	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.48	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
ifDescr.1	TEXT	R/O	StringRange R/O DIS
ifDescr.2	TEXT	R/O	StringRange R/O DIS
ifDescr.3	TEXT	R/O	StringRange R/O DIS
ifDescr.4	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifDescr.5	TEXT	R/O	StringRange R/O DIS
ifDescr.6	TEXT	R/O	StringRange R/O DIS
ifDescr.7	TEXT	R/O	StringRange R/O DIS
ifDescr.8	TEXT	R/O	StringRange R/O DIS
ifDescr.9	TEXT	R/O	StringRange R/O DIS
ifDescr.10	TEXT	R/O	StringRange R/O DIS
ifDescr.11	TEXT	R/O	StringRange R/O DIS
ifDescr.12	TEXT	R/O	StringRange R/O DIS
ifDescr.13	TEXT	R/O	StringRange R/O DIS
ifDescr.14	TEXT	R/O	StringRange R/O DIS
ifDescr.15	TEXT	R/O	StringRange R/O DIS
ifDescr.16	TEXT	R/O	StringRange R/O DIS
ifDescr.17	TEXT	R/O	StringRange R/O DIS
ifDescr.18	TEXT	R/O	StringRange R/O DIS
ifDescr.19	TEXT	R/O	StringRange R/O DIS
ifDescr.20	TEXT	R/O	StringRange R/O DIS
ifDescr.21	TEXT	R/O	StringRange R/O DIS
ifDescr.22	TEXT	R/O	StringRange R/O DIS
ifDescr.23	TEXT	R/O	StringRange R/O DIS
ifDescr.24	TEXT	R/O	StringRange R/O DIS
ifDescr.25	TEXT	R/O	StringRange R/O DIS
ifDescr.26	TEXT	R/O	StringRange R/O DIS
ifDescr.27	TEXT	R/O	StringRange R/O DIS
ifDescr.28	TEXT	R/O	StringRange R/O DIS
ifDescr.29	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifDescr.30	TEXT	R/O	StringRange R/O DIS
ifDescr.31	TEXT	R/O	StringRange R/O DIS
ifDescr.32	TEXT	R/O	StringRange R/O DIS
ifDescr.33	TEXT	R/O	StringRange R/O DIS
ifDescr.34	TEXT	R/O	StringRange R/O DIS
ifDescr.35	TEXT	R/O	StringRange R/O DIS
ifDescr.36	TEXT	R/O	StringRange R/O DIS
ifDescr.37	TEXT	R/O	StringRange R/O DIS
ifDescr.38	TEXT	R/O	StringRange R/O DIS
ifDescr.39	TEXT	R/O	StringRange R/O DIS
ifDescr.40	TEXT	R/O	StringRange R/O DIS
ifDescr.41	TEXT	R/O	StringRange R/O DIS
ifDescr.42	TEXT	R/O	StringRange R/O DIS
ifDescr.43	TEXT	R/O	StringRange R/O DIS
ifDescr.44	TEXT	R/O	StringRange R/O DIS
ifDescr.45	TEXT	R/O	StringRange R/O DIS
ifDescr.46	TEXT	R/O	StringRange R/O DIS
ifDescr.47	TEXT	R/O	StringRange R/O DIS
ifDescr.48	TEXT	R/O	StringRange R/O DIS
ifOperStatus.1	TEXT	R/O	StringRange R/O DIS
ifOperStatus.2	TEXT	R/O	StringRange R/O DIS
ifOperStatus.3	TEXT	R/O	StringRange R/O DIS
ifOperStatus.4	TEXT	R/O	StringRange R/O DIS
ifOperStatus.5	TEXT	R/O	StringRange R/O DIS
ifOperStatus.6	TEXT	R/O	StringRange R/O DIS
ifOperStatus.7	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifOperStatus.8	TEXT	R/O	StringRange R/O DIS
ifOperStatus.9	TEXT	R/O	StringRange R/O DIS
ifOperStatus.10	TEXT	R/O	StringRange R/O DIS
ifOperStatus.11	TEXT	R/O	StringRange R/O DIS
ifOperStatus.12	TEXT	R/O	StringRange R/O DIS
ifOperStatus.13	TEXT	R/O	StringRange R/O DIS
ifOperStatus.14	TEXT	R/O	StringRange R/O DIS
ifOperStatus.15	TEXT	R/O	StringRange R/O DIS
ifOperStatus.16	TEXT	R/O	StringRange R/O DIS
ifOperStatus.17	TEXT	R/O	StringRange R/O DIS
ifOperStatus.18	TEXT	R/O	StringRange R/O DIS
ifOperStatus.19	TEXT	R/O	StringRange R/O DIS
ifOperStatus.20	TEXT	R/O	StringRange R/O DIS
ifOperStatus.21	TEXT	R/O	StringRange R/O DIS
ifOperStatus.22	TEXT	R/O	StringRange R/O DIS
ifOperStatus.23	TEXT	R/O	StringRange R/O DIS
ifOperStatus.24	TEXT	R/O	StringRange R/O DIS
ifOperStatus.25	TEXT	R/O	StringRange R/O DIS
ifOperStatus.26	TEXT	R/O	StringRange R/O DIS
ifOperStatus.27	TEXT	R/O	StringRange R/O DIS
ifOperStatus.28	TEXT	R/O	StringRange R/O DIS
ifOperStatus.29	TEXT	R/O	StringRange R/O DIS
ifOperStatus.30	TEXT	R/O	StringRange R/O DIS
ifOperStatus.31	TEXT	R/O	StringRange R/O DIS
ifOperStatus.32	TEXT	R/O	StringRange R/O DIS
ifOperStatus.33	TEXT	R/O	StringRange R/O DIS
ifOperStatus.34	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifOperStatus.35	TEXT	R/O	StringRange R/O DIS
ifOperStatus.36	TEXT	R/O	StringRange R/O DIS
ifOperStatus.37	TEXT	R/O	StringRange R/O DIS
ifOperStatus.38	TEXT	R/O	StringRange R/O DIS
ifOperStatus.39	TEXT	R/O	StringRange R/O DIS
ifOperStatus.40	TEXT	R/O	StringRange R/O DIS
ifOperStatus.41	TEXT	R/O	StringRange R/O DIS
ifOperStatus.42	TEXT	R/O	StringRange R/O DIS
ifOperStatus.43	TEXT	R/O	StringRange R/O DIS
ifOperStatus.44	TEXT	R/O	StringRange R/O DIS
ifOperStatus.45	TEXT	R/O	StringRange R/O DIS
ifOperStatus.46	TEXT	R/O	StringRange R/O DIS
ifOperStatus.47	TEXT	R/O	StringRange R/O DIS
ifOperStatus.48	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Ifc01 status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ifc02 status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Ifc03 status)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ifc04 status)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ifc05 status)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Ifc06 status)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Ifc07 status)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Ifc08 status)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Ifc09 status)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ifc10 status)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Ifc11 status)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ifc12 status)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Ifc13 status)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Ifc14 status)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Ifc15 status)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Ifc16 status)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Ifc17 status)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Ifc18 status)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Ifc19 status)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Ifc20 status)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Ifc21 status)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Ifc22 status)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Ifc23 status)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Ifc24 status)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Ifc25 status)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Ifc26 status)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Ifc27 status)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Ifc28 status)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Ifc29 status)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Ifc30 status)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Ifc31 status)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Ifc32 status)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Ifc33 status)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Ifc34 status)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Ifc35 status)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Ifc36 status)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Ifc37 status)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Ifc38 status)

name	*type*	*flags*	*range*
faults.39	ALARM	R/O	AlarmFlagRange R/O (Ifc39 status)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Ifc40 status)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Ifc41 status)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Ifc42 status)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Ifc43 status)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Ifc44 status)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Ifc45 status)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Ifc46 status)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Ifc47 status)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Ifc48 status)

9.342 MIB2-Storage-Monitor

The *MIB2-Storage-Monitor* device driver monitors the state of up to 8 entries of a SNMP device which implements the mib-2.hr storage table.

The driver permits to configure the list of entries to be monitored. For this, the table index of each entry to monitor must be entered at the maintenance/setup page of the device window.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational state of each storage monitored.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the redundancy switch in the sat-nms software.

parameter	description
address	The SNMP address specification for the device (details below).
memIndex.1...8	Defines the index of the storage entry which should be monitored
memUnit.1...8	Defines the units the storage should be displayed
memthreshold.1...8	Defines the storage usage in percent which raises an alarm for the respective threshold if exceeded

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` and `private` as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysDescription	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.sysLocation	TEXT	R/O	StringRange R/O
info.sysContact	TEXT	R/O	StringRange R/O
config.memTotalIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memBufferIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memCacheIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memSharedIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.1	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.2	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.memUnit.3	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.4	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.5	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.6	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.7	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.7	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.7	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.8	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.8	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)

name	*type*	*flags*	*range*
config.memThreshold.8	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.9	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.9	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.9	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.10	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.11	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.12	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.13	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)

name	*type*	*flags*	*range*
config.memIndex.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.14	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.15	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.memIndex.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memUnit.16	CHOICE	SAVE SETUP	EnumRange (kByte MByte GByte TByte KiB MiB GiB TiB)
config.memThreshold.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
mem.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.cache	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.buffer	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.shared	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.total.allocUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.cache.allocUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.buffer.allocUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.shared.allocUnits	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.available	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.01.descr	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
mem.01.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.01.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.01.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.01.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.02.descr	TEXT	R/O	StringRange R/O DIS
mem.02.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.02.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.02.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.02.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.03.descr	TEXT	R/O	StringRange R/O DIS
mem.03.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.03.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.03.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.03.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.04.descr	TEXT	R/O	StringRange R/O DIS
mem.04.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.04.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.04.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.04.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.05.descr	TEXT	R/O	StringRange R/O DIS
mem.05.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.05.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.05.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.05.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.06.descr	TEXT	R/O	StringRange R/O DIS
mem.06.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
mem.06.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.06.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.06.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.07.descr	TEXT	R/O	StringRange R/O DIS
mem.07.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.07.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.07.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.07.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.08.descr	TEXT	R/O	StringRange R/O DIS
mem.08.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.08.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.08.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.08.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.09.descr	TEXT	R/O	StringRange R/O DIS
mem.09.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.09.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.09.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.09.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.10.descr	TEXT	R/O	StringRange R/O DIS
mem.10.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.10.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.10.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.10.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.11.descr	TEXT	R/O	StringRange R/O DIS
mem.11.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.11.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.11.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
mem.11.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.12.descr	TEXT	R/O	StringRange R/O DIS
mem.12.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.12.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.12.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.12.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.13.descr	TEXT	R/O	StringRange R/O DIS
mem.13.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.13.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.13.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.13.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.14.descr	TEXT	R/O	StringRange R/O DIS
mem.14.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.14.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.14.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.14.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.15.descr	TEXT	R/O	StringRange R/O DIS
mem.15.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.15.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.15.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.15.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.16.descr	TEXT	R/O	StringRange R/O DIS
mem.16.allocUnits	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
mem.16.total	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
mem.16.used	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
mem.16.usage	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
internal.total	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mem	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.total.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.used.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.cache.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.buffer.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.shared.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.available.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.01.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.02.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.03.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.04.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.05.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.06.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.07.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.08.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.09.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.10.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.11.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.12.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.13.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.14.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.15.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.16.units	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Storage 1 usage)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Storage 2 usage)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Storage 3 usage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Storage 4 usage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Storage 5 usage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Storage 6 usage)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Storage 7 usage)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Storage 8 usage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Storage 9 usage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Storage 10 usage)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Storage 11 usage)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Storage 12 usage)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Storage 13 usage)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Storage 14 usage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Storage 15 usage)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Storage 16 usage)

name	*type*	*flags*	*range*
faults.40	ALARM	R/O	AlarmFlagRange R/O (RAM usage)

9.343 MIB2-UPS-Monitor

This page describes the device driver and the device window for all standard MIB-2 compatible UPS models.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The monitoring parameters
- --- Details for the input lines
- --- Details for the output lines

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.manufacturer	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.upsVersion	TEXT	R/O	StringRange R/O
info.agentVersion	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.attachedDevices	TEXT	R/O	StringRange R/O
info.inputLines	TEXT	R/O	StringRange R/O
info.outputLines	TEXT	R/O	StringRange R/O
config.powerCapacity	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
battery.state	TEXT	R/O	StringRange R/O
battery.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
battery.capacity	INTEGER	R/O	IntegerRange R/O (0 .. 100)
battery.since	INTEGER	R/O	IntegerRange R/O (0 .. 0)
battery.remaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.lineBads	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.1.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.1.voltage	FLOAT	R/O	DoubleRange R/O (0 .. 0)
input.1.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.1.power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.2.frequency	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.2.voltage	FLOAT	R/O	DoubleRange R/O DIS (0 .. 0)
input.2.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.2.power	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
input.3.frequency	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.3.voltage	FLOAT	R/O	DoubleRange R/O DIS (0 .. 0)
input.3.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
input.3.power	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
input.totalPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.source	TEXT	R/O	StringRange R/O
output.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.1.voltage	FLOAT	R/O	DoubleRange R/O (0 .. 0)
output.1.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.1.power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.1.load	INTEGER	R/O	IntegerRange R/O (0 .. 100)
output.2.voltage	FLOAT	R/O	DoubleRange R/O DIS (0 .. 0)
output.2.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.2.power	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
output.2.load	INTEGER	R/O	IntegerRange R/O DIS (0 .. 100)
output.3.voltage	FLOAT	R/O	DoubleRange R/O DIS (0 .. 0)
output.3.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
output.3.power	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
output.3.load	INTEGER	R/O	IntegerRange R/O DIS (0 .. 100)
output.totalPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.totalLoad	INTEGER	R/O	IntegerRange R/O (0 .. 100)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (UPS Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Mains input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low battery)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Depleted battery)

9.344 Macro-Player

The Macro-Player logical device lets you play a macro automatically if a parameter changes or id another device send a message to the macro player device. Several operation modes are available which allow different types of sources which trigger the macro replay.

The Macro-Player is intended to be used in a M&C environment only. It expects the macro to play in the './macros' directory locally on the VLC/M&C. NMS systems store macro files centrally on the NMS server where they not accessible to Macro-Player device.

Macro execution modes

The Macro-Player device may be operated in a number of different modes, selected by the 'execMode' configuration parameter (see 'Configuration parameters' below). The operation mode is used to adapt the device to the type of event it shall use as a trigger for the macro replay:

- EXEC-CONFIGURED --- In this mode the device plays the macro with the name given in the configuration parameter 'macroName' whenever it receives a value for 'execute' variable, regardless of the value received with this message. This mode may be used to make the device respond to a button press from the GUI.
- EXEC-BY-NAME --- In this mode the device plays a macro with the value received with a message to the 'execute' variable. The same device may play multiple macros this way, for instance acting as a common backend for multiple buttons starting one macro each.
- EXEC-IF-EQUAL --- In this mode the device watches a parameter of another device, addressed by the configuration variable 'parameterId'. If the value of the parameter changes to the value as defined in the 'referenceValue' configuration parameter, the macro with the name given in the configuration parameter 'macroName' is played.
- EXEC-IF-NOT-EQUAL --- In this mode the device watches a parameter of another device, addressed by the configuration variable 'parameterId'. If the value of the parameter changes to a value different to the value in the 'referenceValue' configuration parameter, the macro with the name given in the configuration parameter 'macroName' is played.
- EXEC-BY-PARAMETER --- The last mode also watches the variable addressed by the 'parameterId' configuration variable. Any time this variable changes to a new value, this value is treated as a macro name and the macro is played.
- EXEC-BY-TIMER --- Like with EXEC-CONFIGURED, the device plays the macro with the name given in the configuration parameter 'macroName'. In EXEC-BY-TIMER mode the

macro plays automatically every 'interval' seconds, as long as 'timer mode' is set to RUNNING. If 'max count' is set to a non zero value, automatic playing stops after the given number of executions.Using macros with %wait directives together with short interval times may cause the Macro-Player device to start a new instance of the macro before the previous one has finished. While this causes no trouble for the software (the macros will play autonomously in parallel), the result might not be as you expect.

Macros are played in a separate software threads, hence %wait directives in macros may be used freely without delaying other parts of the software. If a macro to be played does not exist, this is silently ignored, also lines within a macro definition which cannot be parsed also are skipped without notice. This is the same behaviour as with macros played from the GUI.

In the modes EXEC-IF-EQUAL, EXEC-IF-NOT-EQUAL and EXEC-BY-PARAMETER the initial value of the watches parameter never starts a macro play. A three cycles startup delay inhibits macro plays during the initialization phase.

In the modes EXEC-IF-EQUAL and EXEC-IF-NOT-EQUAL the 'execute' parameter behaves like in EXEC-CONFIGURED. This means, the configured macro may be triggered by setting 'execute' to an arbitrary value in addition to the configured parameter comparison.

The modes EXEC-IF-EQUAL and EXEC-IF-NOT-EQUAL do a string-based comparison of the unformatted value of the watched parameter. For this reason you should not use FLOAT type parameters for this, because it's very likely that the comparison fails due to rounding inaccuracies. TEXT, CHOICE and INTEGER parameters are safe to use, BOOL or ALARM type parameters can be compared to true/false (all lower case letters).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which configure the behaviour of the device.

parameter	***description***
execMode	The general operation mode of the device as explained above.
macroName	The name of the macro to be played in the modes EXEC-CONFIGURED, EXEC-IF-EQUAL and EXEC-IF-NOT-EQUAL

parameter	*description*
parameterId	The message ID of the parameter to be watched in the modes EXEC-IF-EQUAL, EXEC-IF-NOT-EQUAL and EXEC-BY-PARAMETER
referenceValue	The reference used for the value comparison in the modes EXEC-IF-EQUAL and EXEC-IF-NOT-EQUAL.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
info.nextExecution	TEXT	R/O	StringRange R/O
info.executionCount	TEXT	R/O	StringRange R/O
config.execMode	CHOICE	SAVE SETUP	EnumRange (EXEC-CONFIGURED EXEC-BY-NAME EXEC-IF-EQUAL EXEC-IF-NOT-EQUAL EXEC-BY-PARAMETER EXEC-BY-TIMER)
config.macroName	TEXT	SAVE SETUP	StringRange
config.parameterId	TEXT	SAVE SETUP	StringRange
config.referenceValue	TEXT	SAVE SETUP	StringRange
execute	TEXT		StringRange
interval	INTEGER	SAVE	IntegerRange (0 .. 0)
maxCount	INTEGER	SAVE	IntegerRange (0 .. 0)
timerMode	CHOICE	SAVE	EnumRange (STOPPED RUNNING)

9.345 Matrix-Protection-Switcher

This page describes the device driver and the device window for the*Matrix-Protection-Switcher* logical device. This device lets you use a switch matrix to select the input signal(s) for the backup signal chain in a N:1 redundant configuration controlled by a Protection-Switch-N-To-1

device.

The *Matrix-Protection-Switcher* provides a 'position' parameter which may be commanded to *OFF, 1, 2, 3 ... n*. Each numbered selection routes the input signal(s) from this chain to the backup chain. *OFF* switches the signal at the input off.

The *Matrix-Protection-Switcher* does not act like a simple N:1 switch, it is aware of the type of switch matrix used (distributive, combining ot crosspoint based), evaluates which signals the matrix actually routes to the signal chain to substitute and replicates this matrix settings for the output connected to the backup chain.

Distributive matrix switching behavior

A distributive matrix permits each output to select exactly one input signal. When the *Matrix-Protection-Switcher* switches to position *n* with such a matrix, it looks up which input signal is set for the output connected to chain *n* and replicates this setting for the output to the backup chain. When switching to *OFF*, the *Matrix-Protection-Switcher* simply switches off the signal to the backup chain.

Crosspoint based matrix switching behavior

A crosspoint based switch matrix permits to select an arbitrary combination of input signals for every output. The output signal is the sum of all input signals routed to it. When the *Matrix-Protection-Switcher* switches to position *n* with such a matrix, it looks up which crosspoints are set for the output connected to chain *n* and replicates this setting for the output to the backup chain. When switching to *OFF*, the *Matrix-Protection-Switcher* switches off all crosspoints for the output connected to the backup chain.

Combining matrix switching behavior

A combining matrix permits each input to select exactly one output to route the signal to. The output signal is the sum of all input signals routed to it. When the *Matrix-Protection-Switcher* switches with such a matrix to position *n*, it looks up which inputs are actually routed to chain*n* and switches these inputs to the backup chain. Chain *n* is without input signals after this. When switching to *OFF*, the *Matrix-Protection-Switcher* switches off all inputs which actually are connected to the backup chain.

The combining matrix looses the information which inputs were routed to to the operational chain after switching them to the backup chain. For this reason switching to chain *n* is a one shot operation with this matrix type. In case of a fault in one of the operational chains its signals are switched by the Protection-Switch-N-To-1 device to the backup chain. Switching back however must be done manually by commanding the switch matrix directly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch position.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
matrixName	The name of the switch matrix device to use.
matrixOutputs	A comma separated list of matrix output indexes. The first index denotes the matrix output connected to the backup chain, the second one the output connected to the operational signal chain 1 and so on. The index value must be written exactly in the same way as they appear in the parameter names provided by the matrix device (typically 2digits with leading zero). Example: `05,06,07` defines that the backup signal chain is connected to the matrix output *05*, chain 1 is connected to output *06* and chain 2 to output *07*. The *Matrix-Protection-Switcher* will provide the position settings *OFF, 1, and 2* in this configuration.

Remarks

1. The *Matrix-Protection-Switcher* device evaluates the switch matrix type from the naming of the input/output parameters used by the device driver of the matrix:

- If the driver defines a parameter *output.01* or *output.001* the matrix is considered to be distributive.
- If the driver defines a parameter *input.01* or *input.001* the matrix is considered to be combining.
- If the driver defines a parameter *out.01.in.01* the matrix is considered to be crosspoint based.

2. Most matrix drivers use *01,02,...64* as port index, but matrices with more than 99 ports use 3 digit index values.
3. When using a combining matrix, switching the input signals from chain *n* to the backup chain means that the operational chain has no input signal after this. As the Protection-Switch-N-To-1 device first switches the input signal, then configures the backup equipment and finally switches the backup chain signal on air, there will be some seconds of signal outage between input / output switching.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.matrixType	TEXT	R/O	StringRange R/O
info.matrixInputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.matrixOutputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.matrixName	TEXT	SAVE SETUP	StringRange
config.matrixType	CHOICE	SAVE SETUP	EnumRange (DISTRIBUTIVE COMBINING CROSSPOINT)
config.matrixOutputs	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix summary)

9.346 Matrix-Site-Switch

The Matrix-Site-Switch device supports an antenna 1:1 redundancy setup by controlling a number matrix switch in case of a redundancy switchover. The idea behind this device is simple:

Assume the ports of both antennas are routed to inputs of the switch matrix. appliances like IRDs are connected to the matrix outputs. They are routed to get a signal from one of the antennas. When an antenna switchover happens, all appliances connected to a particular port of the (faulty) antenna shall be switched to the corresponding port of the backup antenna.

Input correspondence lists

For this the device maintains lists of corresponding inputs. There are two of these list, one for switching from primary to backup and one for the opposite direction. The lists contain pairs of input settings which address the corresponding signals on both antennas.

The first list is the 'primaryToBackup' list, it contains an arbitrary number of elements, separated by comma characters. Each element has the form 'XX:YY' where 'XX' is the 2-digit input number selecting the signal from the primary antenna 'YY' is the input number which selects the same signal from the backup antenna. In case of a switchover, the device checks the settings of all outputs addressed by this device if they match one of the 'XX' settings in the list and switches such an output to the corresponding 'YY' input.

The second list, the 'backupToPrimary' list, works exactly the same way, but for backup to primary switch over. This second list is required as switch from one antenna to another is not necessarily invertible.

ETL NGM30 input redundancy

The sat-nms device driver for the ETL NGM30 Matrix is capable to do a redundancy switching for inputs, triggered by the fault flags of the individual matrix inputs. The Matrix-Site-Switch device allows for this redundancy switching and temporarily inhibits this during a site switchover. For the configuration of the Matrix-Site-Switch this means, that both redundant inputs must be mapped to the *primary* input at the other antenna. This gives the protection switching built into the ETL NGM30 Matrix driver to switch to the redundant signal after the site switch if the matrix input reads a fault.

Example:

input	***signal***	***input redundancy***	***antenna redundancy***
01	11L	primary input	primary antenna
02	11L	backup input	primary antenna
03	11H	primary input	primary antenna
04	11H	backup input	primary antenna
05	12L	primary input	primary antenna
06	12L	backup input	primary antenna
07	12H	primary input	primary antenna
08	12H	backup input	primary antenna
09	11L	primary input	backup antenna
10	11L	backup input	backup antenna
11	11H	primary input	backup antenna
12	11H	backup input	backup antenna
13	12L	primary input	backup antenna
14	12L	backup input	backup antenna
15	12H	primary input	backup antenna
16	12H	backup input	backup antenna

The translation tables for the above configuration would be:

primaryToBackup: 01:09,02:09,03:11,04:11, ... 07:15,08:15 backupToPrimary: 09:01,10:01,11:03,12:03, ... 15:07,16:07

Always there are both, the primary and the backup input of the 'from' antenna mapped to the primary input for this signal at the 'to' antenna.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual primary/backup setting as detected by the device
- --- The faults page
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Matrix-Site-Switch device with the devices it shall monitor and control.

parameter	***description***
locationId	The message ID of the parameter to monitor in order to detect the switchover between the primary and the backup antenna.
primaryValue	The value expected from the parameter addressed by 'locationId' when the primary antenna is selected.
backupValue	The value expected from the parameter addressed by 'locationId' when the backup antenna is selected.
matrixName	The device name of the switch matrix to control. Although this device has been developed to work specifically with an ETL-NGM30 switch matrix, it can be used with any switch matrix driver which provides 'output.01' ... 'output.nn' parameter to select the signals routes to each output and an 'input.outputs' parameter which reports the number of outputs available with this particular device.
primaryToBackup	Must be configured to the primary to backup input match list. The format is 'XX:YY,XX:YY,XX:YY,...XX:YY', a comma separated of pairs of input numbers. When switching from primary to backup, all outputs set to an XX input in this list are switched to the corresponding YY input.

parameter	*description*
backupToPrimary	Must be configured to the backup to primary input match list. The format is 'XX:YY,XX:YY,XX:YY,...XX:YY', a comma separated of pairs of input numbers. When switching from backup to primary, all outputs set to an XX input in this list are switched to the corresponding YY input.
outputList	The list of outputs of the switch matrix which shall be controlled by this device. The format is 'AA,BB,CC, ... ,ZZ', listing the 2-digit output numbers to control in a comma separated list. 'outputList' may be left empty, the device then controls all outputs of the switch matrix.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.matrixMode	CHOICE	SAVE SETUP	EnumRange (DISTRIBUTIVE COMBINING)
config.matrixName	TEXT	SAVE SETUP	StringRange
config.locationId	TEXT	SAVE SETUP	StringRange
config.primaryValue	TEXT	SAVE SETUP	StringRange
config.backupValue	TEXT	SAVE SETUP	StringRange
config.primaryToBackup	TEXT	SAVE SETUP	StringRange
config.backupToPrimary	TEXT	SAVE SETUP	StringRange
config.outputList	TEXT	SAVE SETUP	StringRange
config.inputList	TEXT	SAVE SETUP	StringRange
forceSwitch	CHOICE	NOPRESET	EnumRange (PRIMARY BACKUP)

name	*type*	*flags*	*range*
position	TEXT	R/O	StringRange R/O

9.347 Maxtech-LRC-1200

This page describes the device driver and the device window for the Maxtech LRC-1200 1:2 LNA/LNB redundancy controller. The command set is the same as for Vertex-RSC-1200 but the message body contains an asterisk as last byte. The driver was tested with version: **E000 1.27]

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- Meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *19200,8N1* .

Communication protocol / Interface type

This device driver requires the Vertex protocol. Please remember do define a address or use the factory default 0.

Installation

Before you can operate the unit with the satnms software, you must prepare the unit to match the interface parameters expected by the software.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitType	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
protection	CHOICE		EnumRange (MANUAL AUTO)
pol1	CHOICE		EnumRange (MAIN STANDBY)
pol2	CHOICE		EnumRange (MAIN STANDBY)
position	CHOICE	R/O	EnumRange R/O (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)
priority	CHOICE	R/O	EnumRange R/O (NONE POL1 POL2)

name	*type*	*flags*	*range*
revert	CHOICE	R/O	EnumRange R/O (ENABLED DISABLED)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unit 3 (Spare))
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Unit current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Protection)
internal.unitType	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pol1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pol2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.348 Measurement-Converter

The logical device Measurement-Converter translates of a measurement value to a physical quantity in linear context. This is for example useful to converts a measure current (in mA) to humidity, temperature or pressure.

The device calculates gradient (m = delta y / delta x) and y-intercept (n = y - m*x) from to given points configured in the device setup (minIn:maxIn and minOut:maxOut). On every change of the input measurement, a new physical quantity is calculated.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page show the converted value
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
input.Id	Parameter ID of the value which should be converted
minIn	Minimum of input value
maxIn	Maximum of input value
minOut	Minimum of output value
maxOut	Maximum of output value
max	Maximum / upper limit
returnMin	value which will be returned if the input is below min
returnMax	value which will be returned if the input is above max
unit	unit of output value

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.input.Id	TEXT	SAVE SETUP	StringRange
config.minIn	TEXT	SAVE SETUP	StringRange
config.maxIn	TEXT	SAVE SETUP	StringRange
config.minOut	TEXT	SAVE SETUP	StringRange
config.maxOut	TEXT	SAVE SETUP	StringRange
config.unit	CHOICE	SAVE SETUP	EnumRange (mbar % °C ° ppm m/s m3/h kLux)
input	FLOAT		DoubleRange (0.00 .. 0.00)
value	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.m	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.n	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

9.349 Mediakind-CE1

This page describes the device driver and the device window for the Mediakind CE-1 Multiencoder, controlled via REST HTTP.

The driver was tested with release **16.5.0**

The driver can only control one encoding service/instance. As the Mediakind CE-1 can contain more than one "encoder". Each service instance has to be setup in a single device. The respective instance has to be configured at first setup on the device setup page (config.encoder).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general mpeg and ip output parameters.
- --- Contains the Reliable Stream (SRT, RTMP, RIST, ZIXI) In and Output Parameters.
- --- Contains the Reliable Strean In and Output Statistics.
- --- Contains the video settings.

- --- Contains the audio settings.
- --- Contains the conditional access (BISS) parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Mediakind CE-1.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
encoder	Defines the desired encoder service instance.
interfaces	Add the available network interfaces defined on the CE-1 as a comma separated list as these cannot be automatically read from the API. Make sure that you use the exact same naming as on the device.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.encoderName	TEXT	R/O	StringRange R/O
info.transportName	TEXT	R/O	StringRange R/O
info.serverId	TEXT	R/O	StringRange R/O
info.componentOne	TEXT	R/O	StringRange R/O
info.componentTwo	TEXT	R/O	StringRange R/O
config.encoder	CHOICE	SAVE SETUP	EnumRange (?)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.interfaces	TEXT	SAVE SETUP	StringRange
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.channel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.output	TEXT	R/O	StringRange R/O
internal.inputMode	TEXT	R/O	StringRange R/O
internal.inputModeSet	TEXT	R/O	StringRange R/O
internal.outputMode	TEXT	R/O	StringRange R/O
internal.outputModeSet	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.cmd	TEXT	R/O	StringRange R/O
internal.serviceId	TEXT	R/O	StringRange R/O
internal.transportId	TEXT	R/O	StringRange R/O
internal.inputId	TEXT	R/O	StringRange R/O
internal.outputId	TEXT	R/O	StringRange R/O
internal.videoId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.videoTsId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.videoInpId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.audioInp3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioInp8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.videoStatId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioStat8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
actState	TEXT	R/O	StringRange R/O
actStateSrt	TEXT	R/O	StringRange R/O
opState	CHOICE	NOPRESET	EnumRange (START STOP)
opStateSrt	CHOICE	NOPRESET	EnumRange (START STOP)
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
startSrt	TEXT	NOPRESET	StringRange
stopSrt	TEXT	NOPRESET	StringRange
output.provider	TEXT		StringRange
output.service	TEXT		StringRange
output.standard	CHOICE		EnumRange (ATSC DVB)

name	*type*	*flags*	*range*
output.psiPeriod	INTEGER		IntegerRange (20 .. 1000)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.pmtpid	INTEGER		IntegerRange (16 .. 8190)
output.pcrpid	INTEGER		IntegerRange (16 .. 8190)
output.pcrPeriod	INTEGER		IntegerRange (20 .. 1000)
output.gopSignaling	CHOICE		EnumRange (IDR)
enc.ipout.1.protocol	CHOICE		EnumRange (UDP RTP)
enc.ipout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
enc.ipout.1.tos	INTEGER		IntegerRange (0 .. 255)
enc.ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
enc.ipout.1.address	TEXT		StringRange
enc.ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
video.pid	INTEGER		IntegerRange (0 .. 8190)
video.source	TEXT	R/O	StringRange R/O
video.inputLabel	TEXT	R/O	StringRange R/O
video.connector	INTEGER	NOPRESET	IntegerRange (1 .. 8)
video.uhdEnable	CHOICE		EnumRange (OFF ON)
video.codec	CHOICE		EnumRange (H264 HEVC)
video.profile	CHOICE		EnumRange (MAIN HIGH HIGH-10Bit HIGH422-10Bit MAIN MAIN-10Bit MAIN422-10Bit)
video.resolution	CHOICE		EnumRange (UNKOWN 1280x720 1920x1080 3840x2160)
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.rateControl	CHOICE		EnumRange (CBR VBR)

name	*type*	*flags*	*range*
video.datarate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.maxrate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.dynamicRangeMode	CHOICE		EnumRange (SDR FOLLOW INPUT)
video.delayMode	CHOICE		EnumRange (SHORT STANDARD LONG CUSTOM)
video.buffer	INTEGER		IntegerRange (0 .. 0)
video.gopType	CHOICE		EnumRange (AUTO CLOSED OPEN)
video.bFrames	CHOICE		EnumRange (AUTO OFF 1 2 3)
video.keyPeriod	INTEGER		IntegerRange (500 .. 10000)
video.frameRate	TEXT	R/O	StringRange R/O
video.running	BOOLEAN	R/O	BooleanRange R/O (true false)
video.ar	TEXT	R/O	StringRange R/O
video.colorSpace	TEXT	R/O	StringRange R/O
video.dynamicRange	TEXT	R/O	StringRange R/O
video.standard	TEXT	R/O	StringRange R/O
video.bitdepth	TEXT	R/O	StringRange R/O
audio.1.enable	CHOICE		EnumRange (OFF ON)
audio.1.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.1.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.1.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)

name	*type*	*flags*	*range*
audio.1.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.1.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.1.pid	INTEGER		IntegerRange (0 .. 8190)
audio.1.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.enable	CHOICE		EnumRange (OFF ON)
audio.2.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.2.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.2.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)
audio.2.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.2.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.2.pid	INTEGER		IntegerRange (0 .. 8190)
audio.2.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.enable	CHOICE		EnumRange (OFF ON)
audio.3.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.3.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.3.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)

name	*type*	*flags*	*range*
audio.3.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.3.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.3.pid	INTEGER		IntegerRange (0 .. 8190)
audio.3.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.enable	CHOICE		EnumRange (OFF ON)
audio.4.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.4.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.4.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)
audio.4.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.4.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.4.pid	INTEGER		IntegerRange (0 .. 8190)
audio.4.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.enable	CHOICE		EnumRange (OFF ON)
audio.5.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.5.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.5.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)

name	*type*	*flags*	*range*
audio.5.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.5.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.5.pid	INTEGER		IntegerRange (0 .. 8190)
audio.5.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.enable	CHOICE		EnumRange (OFF ON)
audio.6.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.6.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.6.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)
audio.6.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.6.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.6.pid	INTEGER		IntegerRange (0 .. 8190)
audio.6.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.enable	CHOICE		EnumRange (OFF ON)
audio.7.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.7.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.7.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)

name	*type*	*flags*	*range*
audio.7.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.7.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.7.pid	INTEGER		IntegerRange (0 .. 8190)
audio.7.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.enable	CHOICE		EnumRange (OFF ON)
audio.8.coding	CHOICE		EnumRange (MPEG-L2 PASSTHROUGH DOLBY-DIGITAL DOLBY-DIGITAL-PLUS AAC HE-AAC HE-AAC-V2)
audio.8.datarate	INTEGER		IntegerRange (32 .. 1024)
audio.8.mode	CHOICE		EnumRange (MONO MONO-LEFT MONO-RIGHT DUAL-MONO STEREO)
audio.8.samplingRate	CHOICE		EnumRange (32kHz 44.1KHz 48kHz)
audio.8.aacProfile	CHOICE		EnumRange (HIGH-QUALITY AAC)
audio.8.pid	INTEGER		IntegerRange (0 .. 8190)
audio.8.levelLeft	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.levelRight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.mode	CHOICE		EnumRange (SRT IP ZIXI RIST RTMP)
ipin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 0)
ipin.1.rtpHeader	CHOICE		EnumRange (ON OFF)
srtin.1.mode	CHOICE		EnumRange (CALLER LISTENER)

name	*type*	*flags*	*range*
srtin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
srtin.1.address	TEXT		StringRange
srtin.1.port	INTEGER		IntegerRange (0 .. 65535)
srtin.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srtin.1.latency	INTEGER		IntegerRange (0 .. 6000)
srtin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
srtin.1.passphrase	TEXT	NOPRESET	StringRange
srtin.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtin.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
srtin.1.fileTransfer	CHOICE		EnumRange (OFF ON)
srtin.1.maxSize	INTEGER		IntegerRange (0 .. 0)
zixiin.1.remote	TEXT		StringRange
zixiin.1.address	TEXT		StringRange
zixiin.1.stream	TEXT		StringRange
zixiin.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixiin.1.passphrase	TEXT	NOPRESET	StringRange
zixiin.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixiin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
ristin.1.address	TEXT		StringRange
ristin.1.port	INTEGER		IntegerRange (1024 .. 65534)
ristin.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
ristin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
ristin.1.passphrase	TEXT	NOPRESET	StringRange
ristin.1.minRtt	INTEGER		IntegerRange (0 .. 0)
ristin.1.maxRtt	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ristin.1.minRetries	INTEGER		IntegerRange (0 .. 0)
ristin.1.maxRetries	INTEGER		IntegerRange (0 .. 0)
ristin.1.returnBw	INTEGER		IntegerRange (0 .. 0)
ristin.1.peer	TEXT		StringRange
rtmpin.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmpin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
rtmpin.1.address	TEXT		StringRange
rtmpin.1.port	INTEGER		IntegerRange (1024 .. 65534)
rtmpin.1.stream	TEXT		StringRange
rtmpin.1.application	TEXT		StringRange
rtmpin.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
rtmpin.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ipout.1.mode	CHOICE		EnumRange (SRT IP ZIXI RIST RTMP)
ipout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (1024 .. 65534)
ipout.1.sourceAddress	TEXT		StringRange
ipout.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
ipout.1.rtpHeader	CHOICE		EnumRange (OFF ON)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
srtout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
srtout.1.address	TEXT		StringRange
srtout.1.port	INTEGER		IntegerRange (0 .. 65535)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
srtout.1.latency	INTEGER		IntegerRange (60 .. 8000)
srtout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
srtout.1.passphrase	TEXT	NOPRESET	StringRange
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
srtout.1.fileTransfer	CHOICE		EnumRange (OFF ON)
srtout.1.maxSize	INTEGER		IntegerRange (0 .. 0)
zixiout.1.address	TEXT		StringRange
zixiout.1.stream	TEXT		StringRange
zixiout.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixiout.1.passphrase	TEXT	NOPRESET	StringRange
zixiout.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixiout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
zixiout.1.datarate	INTEGER		IntegerRange (0 .. 0)
ristout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ristout.1.port	INTEGER		IntegerRange (1024 .. 65534)
ristout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ristout.1.returnBw	INTEGER		IntegerRange (0 .. 0)
ristout.1.minBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.maxBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.reorderBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
ristout.1.passphrase	TEXT	NOPRESET	StringRange
ristout.1.minRtt	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ristout.1.maxRtt	INTEGER		IntegerRange (0 .. 0)
ristout.1.minRetries	INTEGER		IntegerRange (0 .. 0)
ristout.1.maxRetries	INTEGER		IntegerRange (0 .. 0)
ristout.1.bw	INTEGER		IntegerRange (0 .. 0)
ristout.1.peer	TEXT		StringRange
rtmpout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmpout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
rtmpout.1.address	TEXT		StringRange
rtmpout.1.port	INTEGER		IntegerRange (1024 .. 65534)
rtmpout.1.stream	TEXT		StringRange
rtmpout.1.application	TEXT		StringRange
rtmpout.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
rtmpout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ipin.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.available.bufferRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.bufferTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.windowSizeCong	TEXT	R/O	StringRange R/O
ipin.1.windowSizeFlow	TEXT	R/O	StringRange R/O
ipin.1.meas.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.rtt	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipin.1.bufferRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.mtu	TEXT	R/O	StringRange R/O
ipin.1.delayTx	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.1.delayRx	TEXT	R/O	StringRange R/O
ipin.1.period	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsFlight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.remote	TEXT	R/O	StringRange R/O
ipin.1.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.bandwidthCeiling	TEXT	R/O	StringRange R/O
ipin.1.packetsNotAck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.timespanNotAck	TEXT	R/O	StringRange R/O
ipin.1.packetsUndelivered	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.noClients	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.datarate	TEXT	R/O	StringRange R/O
ipout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.available.bufferTx	TEXT	R/O	StringRange R/O
ipout.1.windowSizeCong	TEXT	R/O	StringRange R/O
ipout.1.windowSizeFlow	TEXT	R/O	StringRange R/O
ipout.1.meas.bandwidth	TEXT	R/O	StringRange R/O
ipout.1.rtt	TEXT	R/O	StringRange R/O
ipout.1.bufferTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.mtu	TEXT	R/O	StringRange R/O
ipout.1.delayTx	TEXT	R/O	StringRange R/O
ipout.1.delayRx	TEXT	R/O	StringRange R/O
ipout.1.period	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipout.1.packetsFlight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.remote	TEXT	R/O	StringRange R/O
ipout.1.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.bandwidthCeiling	TEXT	R/O	StringRange R/O
ipout.1.packetsNotAck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.timespaceUndelivered	TEXT	R/O	StringRange R/O
ipout.1.packetsUndelivered	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.noClients	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video input)

9.350 Mediakind-RX1

This page describes the device driver and the device window for the Mediakind RX-1 Multidecoder, controlled via REST HTTP.

The driver was tested with release**16.5.1.2**

The driver can only control one decoding service/instance. As the Mediakind RX-1 can contain more than one "decoder". Each service instance has to be setup in a single device. The respective instance has to be configured at first setup on the device setup page (config.decoder).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the Main Input, ASI and Program Selection.
- --- Contains the RF and IP Input Parameters
- --- Contains the Reliable Stream (SRT, RTMP, RIST, ZIXI) In and Output Parameters.
- --- Contains the Reliable Strean In and Output Statistics.

- --- Contains the video settings.
- --- Contains the audio settings.
- --- Contains the conditional access (BISS) parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Mediakind RX-1.

parameter	***description***
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lowFreqLim	Define the lower frequency limit if desired. By default the frequency do not contain a valid range
highFreqLim	Define the upper frequency limit if desired. By default the frequency do not contain a valid range
lowSymRateLim	Define the minimum symbolrate if desired. By default the symbolrate do not contain a valid range
highSymRateLim	Define the maximum symbolrate if desired. By default the symbolrate do not contain a valid range
22kHz	Set 22kz state for 1st RF input
lnbVoltage	Set the lnb voltage for 1st RF input
22kHz.2	Set 22kz state for 2nd RF input
lnbVoltage.2	Set the lnb voltage for 2nd RF input
decoder	Defines the desired decoder service instance.
stream	Defines the desired reliable stream service instance.
interfaces	Add the available network interfaces defined on the CE-1 as a comma separated list as these cannot be automatically read from the API. Make sure that you use the exact same naming as on the device.

parameter	*description*
marginThreshold	Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise a "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value is stored as a saved parameter on the MNC rather than as a parameter in the IRD.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

HTTP session reuse

Multiple devices located in the same frame (using the same IP address) can be configured to share the same HTTP session. For this enter the IP address and login credentials at the 'address' config parameter of the first device. For all other devices enter the name of the first device instead. These devices then reuse the HTTP session / connection of the first device. This has several advantadges:

- There is only one HTTP session for all devices and only one login at the physical device. this is specialy helpful for devices which limit the number of session which are opened in parallel like the Haivision-SRT-Gateway.
- HTTP keep-alive will be enabled for all HTTP requests to the device. This reduces the number of TCP sockets opened and (specially with HTTPS) significantly improves performance.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.receiverName	TEXT	R/O	StringRange R/O
info.transportName	TEXT	R/O	StringRange R/O
info.chassisName	TEXT	R/O	StringRange R/O
info.memSize	TEXT	R/O	StringRange R/O
info.diskSize	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.22kHz.2	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage.2	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.decoder	CHOICE	SAVE SETUP	EnumRange (?)
config.stream	CHOICE	SAVE SETUP	EnumRange (?)
config.interfaces	TEXT	SAVE SETUP	StringRange
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.channel	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.stream	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.configVersion	TEXT	R/O	StringRange R/O
internal.postType	TEXT	R/O	StringRange R/O
internal.programNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.cmd	TEXT	R/O	StringRange R/O
internal.lo	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.lo2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.output	TEXT	R/O	StringRange R/O
internal.inputMode	TEXT	R/O	StringRange R/O
internal.inputModeSet	TEXT	R/O	StringRange R/O
internal.outputMode	TEXT	R/O	StringRange R/O
internal.outputModeSet	TEXT	R/O	StringRange R/O
internal.ifc	TEXT	R/O	StringRange R/O
internal.ifc2	TEXT	R/O	StringRange R/O
internal.serviceId	TEXT	R/O	StringRange R/O
internal.serverId	TEXT	R/O	StringRange R/O
internal.inputId	TEXT	R/O	StringRange R/O
internal.outputId	TEXT	R/O	StringRange R/O
internal.transportId	TEXT	R/O	StringRange R/O
internal.videoId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.audio6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audio8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioOut8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.videoTsId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs1Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs2Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs3Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs4Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs5Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs6Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs7Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.audioTs8Id	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nAudios	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
actState	TEXT	R/O	StringRange R/O
actStateSrt	TEXT	R/O	StringRange R/O
opState	CHOICE	NOPRESET	EnumRange (START STOP)
opStateSrt	CHOICE	NOPRESET	EnumRange (START STOP)
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
startSrt	TEXT	NOPRESET	StringRange
stopSrt	TEXT	NOPRESET	StringRange
interface	CHOICE		EnumRange (IP ASI SAT)
interface.2	CHOICE		EnumRange (IP ASI SAT)
delayMode	CHOICE		EnumRange (LOW STANDARD COMPATIBILTY SUPER LOW)
input.redundancy	CHOICE		EnumRange (OFF ACTIVE-PASSIVE ACTIVE-ACTIVE ACTIVE-ACTIVE-SWITCH ST-2022-7)
input.maxSkew	INTEGER		IntegerRange (10 .. 450)
input.lossTimeout	INTEGER		IntegerRange (0 .. 10000)
asi.1.packetSize	CHOICE		EnumRange (188 204)
input	CHOICE	NOPRESET	EnumRange (1 2 3 4)
asi.2.packetSize	CHOICE		EnumRange (188 204)
input.2	CHOICE	NOPRESET	EnumRange (1 2 3 4)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange	INTEGER		IntegerRange (2000 .. 50000)
goldCode	INTEGER		IntegerRange (0 .. 262142)

name	*type*	*flags*	*range*
misFilter	CHOICE		EnumRange (ON OFF)
misStreamId	INTEGER		IntegerRange (0 .. 255)
frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate.2	FLOAT		DoubleRange (0.000 .. 0.000)
searchRange.2	INTEGER		IntegerRange (2000 .. 50000)
goldCode.2	INTEGER		IntegerRange (0 .. 262142)
misFilter.2	CHOICE		EnumRange (ON OFF)
misStreamId.2	INTEGER		IntegerRange (0 .. 255)
dec.ipin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
dec.ipin.1.address	TEXT		StringRange
dec.ipin.1.port	INTEGER		IntegerRange (0 .. 0)
dec.ipin.2.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
dec.ipin.2.address	TEXT		StringRange
dec.ipin.2.port	INTEGER		IntegerRange (0 .. 0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
dvbmode	TEXT	R/O	StringRange R/O
modulation	TEXT	R/O	StringRange R/O
pilots	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
rolloff	TEXT	R/O	StringRange R/O
state.ber	TEXT	R/O	StringRange R/O
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.dvbmode	TEXT	R/O	StringRange R/O
totalBitCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
bitError	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
fecError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ccError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pidError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pmtError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
syncByteError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
transportError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tsSyncLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
serviceMode	CHOICE		EnumRange (AUTO MANUAL)
state.scrambled	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.bissCa	CHOICE		EnumRange (ON OFF)
video.pid	INTEGER		IntegerRange (0 .. 8191)
video.fail	CHOICE		EnumRange (FREEZE BLACK OFF)
video.sdiFormat	CHOICE		EnumRange (NONE 1080i 720p 1080p SD)
video.codec	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
video.connector	INTEGER	NOPRESET	IntegerRange (0 .. 8)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
audio.1.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.1.pid	INTEGER		IntegerRange (0 .. 8190)
audio.1.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.1.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.1.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.channelMode	TEXT	R/O	StringRange R/O
audio.1.language	TEXT	R/O	StringRange R/O
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.2.pid	INTEGER		IntegerRange (0 .. 8190)
audio.2.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.2.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.2.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.2.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.channelMode	TEXT	R/O	StringRange R/O
audio.2.language	TEXT	R/O	StringRange R/O
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.3.pid	INTEGER		IntegerRange (0 .. 8190)
audio.3.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.3.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.3.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.channelMode	TEXT	R/O	StringRange R/O
audio.3.language	TEXT	R/O	StringRange R/O
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.4.pid	INTEGER		IntegerRange (0 .. 8190)
audio.4.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.4.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)

name	*type*	*flags*	*range*
audio.4.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.channelMode	TEXT	R/O	StringRange R/O
audio.4.language	TEXT	R/O	StringRange R/O
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.5.pid	INTEGER		IntegerRange (0 .. 8190)
audio.5.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.5.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.5.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.channelMode	TEXT	R/O	StringRange R/O
audio.5.language	TEXT	R/O	StringRange R/O
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)

name	*type*	*flags*	*range*
audio.6.pid	INTEGER		IntegerRange (0 .. 8190)
audio.6.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.6.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.6.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.channelMode	TEXT	R/O	StringRange R/O
audio.6.language	TEXT	R/O	StringRange R/O
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.7.pid	INTEGER		IntegerRange (0 .. 8190)
audio.7.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.7.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.7.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.channelMode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.7.language	TEXT	R/O	StringRange R/O
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.mode	CHOICE		EnumRange (STEREO 5.1 16 PASSTHROUGH)
audio.8.pid	INTEGER		IntegerRange (0 .. 8190)
audio.8.pidMode	CHOICE	R/O	EnumRange R/O (MANUAL AUTO)
audio.8.lipSyncOffset	FLOAT		DoubleRange (-50.0 .. 50.0)
audio.8.sdiMatrix	CHOICE		EnumRange (OFF EMBEDDED-1 EMBEDDED-2 EMBEDDED-3 EMBEDDED-4 EMBEDDED-5 EMBEDDED-6 EMBEDDED-7 EMBEDDED-8)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.channelMode	TEXT	R/O	StringRange R/O
audio.8.language	TEXT	R/O	StringRange R/O
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.1.mode	CHOICE		EnumRange (SRT IP ZIXI RIST RTMP)
ipin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 0)
ipin.1.rtpHeader	CHOICE		EnumRange (ON OFF)
srtin.1.mode	CHOICE		EnumRange (CALLER LISTENER)
srtin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)

name	*type*	*flags*	*range*
srtin.1.address	TEXT		StringRange
srtin.1.port	INTEGER		IntegerRange (0 .. 65535)
srtin.1.localPort	INTEGER		IntegerRange (0 .. 65535)
srtin.1.latency	INTEGER		IntegerRange (0 .. 6000)
srtin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
srtin.1.passphrase	TEXT	NOPRESET	StringRange
srtin.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtin.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
srtin.1.fileTransfer	CHOICE		EnumRange (OFF ON)
srtin.1.maxSize	INTEGER		IntegerRange (0 .. 0)
zixiin.1.remote	TEXT		StringRange
zixiin.1.address	TEXT		StringRange
zixiin.1.stream	TEXT		StringRange
zixiin.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixiin.1.passphrase	TEXT	NOPRESET	StringRange
zixiin.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixiin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
ristin.1.address	TEXT		StringRange
ristin.1.port	INTEGER		IntegerRange (1024 .. 65534)
ristin.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
ristin.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
ristin.1.passphrase	TEXT	NOPRESET	StringRange
ristin.1.minRtt	INTEGER		IntegerRange (0 .. 0)
ristin.1.maxRtt	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ristin.1.minRetries	INTEGER		IntegerRange (0 .. 0)
ristin.1.maxRetries	INTEGER		IntegerRange (0 .. 0)
ristin.1.returnBw	INTEGER		IntegerRange (0 .. 0)
ristin.1.peer	TEXT		StringRange
rtmpin.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmpin.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
rtmpin.1.address	TEXT		StringRange
rtmpin.1.port	INTEGER		IntegerRange (1024 .. 65534)
rtmpin.1.stream	TEXT		StringRange
rtmpin.1.application	TEXT		StringRange
rtmpin.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
rtmpin.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ipout.1.mode	CHOICE		EnumRange (SRT IP ZIXI RIST RTMP)
ipout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (1024 .. 65534)
ipout.1.sourceAddress	TEXT		StringRange
ipout.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
ipout.1.rtpHeader	CHOICE		EnumRange (OFF ON)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
srtout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
srtout.1.address	TEXT		StringRange
srtout.1.port	INTEGER		IntegerRange (0 .. 65535)
srtout.1.localPort	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
srtout.1.latency	INTEGER		IntegerRange (60 .. 8000)
srtout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
srtout.1.passphrase	TEXT	NOPRESET	StringRange
srtout.1.overheadBw	INTEGER		IntegerRange (5 .. 100)
srtout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
srtout.1.fileTransfer	CHOICE		EnumRange (OFF ON)
srtout.1.maxSize	INTEGER		IntegerRange (0 .. 0)
zixiout.1.address	TEXT		StringRange
zixiout.1.stream	TEXT		StringRange
zixiout.1.port	INTEGER		IntegerRange (1025 .. 65525)
zixiout.1.passphrase	TEXT	NOPRESET	StringRange
zixiout.1.latency	INTEGER		IntegerRange (30 .. 10000)
zixiout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
zixiout.1.datarate	INTEGER		IntegerRange (0 .. 0)
ristout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
ristout.1.port	INTEGER		IntegerRange (1024 .. 65534)
ristout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ristout.1.returnBw	INTEGER		IntegerRange (0 .. 0)
ristout.1.minBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.maxBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.reorderBuffer	INTEGER		IntegerRange (0 .. 0)
ristout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-265)
ristout.1.passphrase	TEXT	NOPRESET	StringRange
ristout.1.minRtt	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ristout.1.maxRtt	INTEGER		IntegerRange (0 .. 0)
ristout.1.minRetries	INTEGER		IntegerRange (0 .. 0)
ristout.1.maxRetries	INTEGER		IntegerRange (0 .. 0)
ristout.1.bw	INTEGER		IntegerRange (0 .. 0)
ristout.1.peer	TEXT		StringRange
rtmpout.1.mode	CHOICE		EnumRange (CALLER LISTENER)
rtmpout.1.interface	CHOICE	NOPRESET	EnumRange (lo eth1 eth2)
rtmpout.1.address	TEXT		StringRange
rtmpout.1.port	INTEGER		IntegerRange (1024 .. 65534)
rtmpout.1.stream	TEXT		StringRange
rtmpout.1.application	TEXT		StringRange
rtmpout.1.sourcePort	INTEGER		IntegerRange (1024 .. 65534)
rtmpout.1.maxConnections	INTEGER		IntegerRange (0 .. 0)
ipin.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.available.bufferRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.bufferTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.windowSizeCong	TEXT	R/O	StringRange R/O
ipin.1.windowSizeFlow	TEXT	R/O	StringRange R/O
ipin.1.meas.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.rtt	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.1.bufferRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.mtu	TEXT	R/O	StringRange R/O
ipin.1.delayTx	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.1.delayRx	TEXT	R/O	StringRange R/O
ipin.1.period	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsFlight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.remote	TEXT	R/O	StringRange R/O
ipin.1.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.bandwidthCeiling	TEXT	R/O	StringRange R/O
ipin.1.packetsNotAck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.timespanNotAck	TEXT	R/O	StringRange R/O
ipin.1.packetsUndelivered	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.quality	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.noClients	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.1.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.available.bufferTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.windowSizeCong	TEXT	R/O	StringRange R/O
ipout.1.windowSizeFlow	TEXT	R/O	StringRange R/O
ipout.1.meas.bandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.1.rtt	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.1.bufferTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.mtu	TEXT	R/O	StringRange R/O
ipout.1.delayTx	TEXT	R/O	StringRange R/O
ipout.1.delayRx	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipout.1.period	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsFlight	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.remote	TEXT	R/O	StringRange R/O
ipout.1.uptime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.bandwidthCeiling	TEXT	R/O	StringRange R/O
ipout.1.packetsNotAck	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.timespaceUndelivered	TEXT	R/O	StringRange R/O
ipout.1.packetsUndelivered	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.1.noClients	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RF Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.351 Meinberg-LantimeM300

This page describes the device driver and the device window for a Meinberg Lantime M300 NTP timeserver with GPS receiver.
The driver is intended for monitoring only purpose.

Please Note: Before operation SNMP has to be enabled from the web interface of the Meinberg.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- GPS position data
- --- NTP data
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.serial.cpu	TEXT	R/O	StringRange R/O
info.ntpVersion	TEXT	R/O	StringRange R/O
info.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
longitude	TEXT	R/O	StringRange R/O
latitude	TEXT	R/O	StringRange R/O
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state	TEXT	R/O	StringRange R/O
ntp.state	TEXT	R/O	StringRange R/O
ntp.stratum	INTEGER	R/O	IntegerRange R/O (0 .. 16)
ntp.refclk	TEXT	R/O	StringRange R/O
ntp.offset	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
ntp.nclients	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.nclientsday	INTEGER	R/O	IntegerRange R/O (0 .. 0)
refclk.type	TEXT	R/O	StringRange R/O
refclk.usage	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (GPS/Ref Clk Sync)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Antenna short circuit)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Antenna disconnected)
faults.04	ALARM	R/O	AlarmFlagRange R/O (NTP Sync)

9.352 Meinberg-SDU

This page describes the device driver and the device window for a Meinberg SDU 10MHz Distribution Amplifier via SNMP.

It was tested with

- SDU-R/SIN/NET/AD10-AD10

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- In and Outputs measurements and main unit settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.supply.1.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.2.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply.3.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.supply.4.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
masterClock	CHOICE		EnumRange (NONE CLOCK 1 CLOCK 2)
accessMode	CHOICE		EnumRange (LOCAL REMOTE)
input.01.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.02.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.01.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.02.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.03.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.04.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.05.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.06.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.07.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.08.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.09.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.10.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.11.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.12.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.13.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.14.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.15.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.16.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
timeDifference	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
protection	CHOICE	R/O	EnumRange R/O (AUTO MANUAL)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Clock 1 Sync)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Clock 2 Sync)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.353 Mesa-WNT

This page describes the device driver and the device window for the*Mesa-WNT* device. This device reads the wind speed and direction values from the Mesa WNT sensor.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual measurement values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	*description*
address	Leave this field empty, the sensor does not require a device address.
Serial	Remember to set the line parameters for the serial interface to*4800,N,8,1* , no handshaking.

Remarks

- You need a version of the sensor which is configured to wait for a request command. It

works not if the sensor sends its values every 3 secondes without a request from the M&C system.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
speed	FLOAT	R/O	DoubleRange R/O (0.0 .. 99.0)
speed2	FLOAT	R/O	DoubleRange R/O (0.0 .. 360.0)
direction	INTEGER	R/O	IntegerRange R/O (0 .. 359)
hiThreshold	FLOAT	SAVE	DoubleRange (0.0 .. 99.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Measuring section)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Speed HI threshold)

9.354 Mesa-WSWD

This page describes the device driver and the device window for the*Mesa-WSWD* device. This device reads the wind speed and direction values from the Mesa WSWD Sonic Anemometer.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Wind measurement.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the RX8200.

parameter	***description***
address	The device address (or ID) as 2 character ASCII string. factory default of WDWD is 00
northOffset	Value for North Correction, changes will be written to EEPROM
Serial	Remember to set the line parameters for the serial interface to*38400,N,8,1* , no handshaking. (default)

Remarks

- You need to configure the sensor to wait for a request command. It works not if the sensor is configured for Free-running Telegram Output.
- It is recommended to use RS422 (4wire) serial connection. Unit should be configured with command 16 to full duplex.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.production	TEXT	R/O	StringRange R/O
info.temp.internal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.temp.transducer	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.northOffset	FLOAT	SETUP	DoubleRange (0.0 .. 360.0)
speed	FLOAT	R/O	DoubleRange R/O (0.0 .. 99.0)
speed2	FLOAT	R/O	DoubleRange R/O (0.0 .. 360.0)
hiThreshold	FLOAT	SAVE	DoubleRange (0.0 .. 99.0)
direction	INTEGER	R/O	IntegerRange R/O (0 .. 359)
windRose	TEXT	R/O	StringRange R/O
heating	CHOICE	R/O	EnumRange R/O (ON OFF)
reset	TEXT	NOPRESET	StringRange
internal.status	HEX	R/O	HexRange R/O (0 .. 0)
internal.dummy	TEXT	R/O	StringRange R/O
faults.02	ALARM	R/O	AlarmFlagRange R/O (High Wind Speed)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (High Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temp. Difference)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Averaging Buffer)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Meas. Track X)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Meas. Track Y)

9.355 MiniCircuits-Switch-ZT

This page describes the device driver and the device window for a Mini-Circuits ZT series custom switch using http (get/post) communications.

The driver is prepared to support MTS, SPDT, SP4T, SP6T and SP8T switches. Therefore the proper switch type has to be selected within the device setup.

If more than one switch is installed on the same controller multiple instances of the device may be configured in the sat-nms M&C to handle all switches. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that HTTP requests to the device do not overlap.

It was tested with the following models:

- Model: ZT-324 MSP6TA-12+
- Firmware: 9-C7-1.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The ZT Series switches needs selection of the correct switch type and number for proper operation.

parameter	***description***
address	The device ip address
tempSensors	Define the number of temperature sensors installed.
password	Define password if http communication is secured using password security.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
swType	Define the switch Type.
swNumber	If more that one switch is installed define the switch number to be controlled.

Remote Control

- TCP uses HTTP11 protocol
- Default TCP/IP port is 80

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.count	TEXT	R/O	StringRange R/O
info.temp0	FLOAT		DoubleRange (0.00 .. 0.00)
info.temp1	FLOAT		DoubleRange (0.00 .. 0.00)
info.temp2	FLOAT		DoubleRange (0.00 .. 0.00)
info.temp3	FLOAT		DoubleRange (0.00 .. 0.00)
info.fan	CHOICE		EnumRange (ON OFF)
config.tempSensors	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4)
config.password	TEXT	SAVE SETUP	StringRange
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.swType	CHOICE	SAVE SETUP	EnumRange (MTS SPDT SP4T SP6T SP8T)
config.swNumber	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
position	CHOICE		EnumRange (0 1 2 3 4 5 6)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
internal.password	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fan 2)

9.356 Mirad-VPC

This page describes the device driver and the device window for the Mirad VPC Switching System a waveguide switch controller and Variable Power Combiner The driver was tested with the 2-way Ka-Band VPC System:

- Project 902.605 for LHCP and RHCP.

Beside device parameter and fault monitoring it allows to set the phase shifter and switch positions as well as starting a reference run.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch control unit
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Remote Control

- Default address A and B
- RS-485 2-Wire

- 9600/8/N/1

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
vpc	CHOICE		EnumRange (UNKNOWN MANUAL 1+1 A 1+1 B COMBINING)
vpcState	CHOICE		EnumRange (FLT A A+B B LOAD)
switch.01	CHOICE		EnumRange (A B)
switch.02	CHOICE		EnumRange (A B)
switch.03	CHOICE		EnumRange (A B)
switch.04	CHOICE		EnumRange (A B)
switch.05	CHOICE		EnumRange (A B)
opMode	CHOICE	R/O	EnumRange R/O (LOCAL LOCAL+REMOTE LOCAL)
detector	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
phasePos	INTEGER		IntegerRange (0 .. 10000)
maxPhasePos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
refRun	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Phase Shifter Ref)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Fail since Ref)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Phase Shifter Moving)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Phase Shifter Low Limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Phase Shifter High Limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch 5)

9.357 MissionMicrowave-SSPA

This page describes the device driver and the device window for the Mission Microwave SSPA and SSPB using SNMP protocol.

It was tested with various Single and Dual Band units using SNMPv3 with the following authentification parameters:

- userrw MD5 bacon123

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters.
- --- The SSPA meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
lowOutputThreshold	Sets the low rf output level threshold level.
lowOutputAlarmType	Sets the low rf output level alarm type.
highOutputThreshold	Sets the high rf output level threshold level.
highOutputAlarmType	Sets the high rf output level alarm type.
highReflThreshold	Sets the high reflected power threshold level.
highReflAlarmType	Sets the high reflected power alarm type.
txReference	Sets the reference source.
referenceFallback	Sets the reference source fallback behaviour.
hasDualBand	Defines if the units is single or dual band.
maxNullResponses	Defines the maximum allowed amount of consecutive Null responses before raising a Communication Fault. This is a Workaround as some units respond with Null on random requests due to unknown/unsolved FW problems.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	**description**
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.

parameter	description
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*

9.358 Mitec-ARK-RX

This page describes the device driver and the device window for the Mitec 1+1 LNB redundancy controller.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The monitoring parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.snmpVersion	TEXT	R/O	StringRange R/O
info.date	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)
reset	TEXT	NOPRESET	StringRange
internal.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (LNB-A Failed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (LNB-A Input Voltage)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (LNB-A PSU)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LNB-A Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LNB-B Failed)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LNB-B Input Voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (LNB-B PSU)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LNB-A Summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Downlink-Switch)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.359 Mitec-Switch

This page describes the device driver and the device window for the Mitec Switching Control Unit 13067-001MD - a 2+1 TWT Redundancy controller and Variable Power Combiner with serial interface.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
reset	TEXT	NOPRESET	StringRange
vpc	CHOICE		EnumRange (FLT A A+B B LOAD)
switch.01	CHOICE		EnumRange (FLT A B)
switch.02	CHOICE		EnumRange (FLT A B)
switch.03	CHOICE		EnumRange (FLT A B)
switch.04	CHOICE		EnumRange (FLT A B)
switch.05	CHOICE		EnumRange (FLT A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Switches)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HPA-1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HPA-2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (HPA-R)

9.360 Mitec-TX

This page describes the device driver and the device window for a Mitec TX Uplink Controller/SSPA Switching Unit.

The driver supports the 1:1 redundancy model and is prepared to support 2:1 redundancy. It is tested with two Advantech SSPAs. Each SSPA is handled as an individual device so therefore 2 or 3 instances of the Mitec-TX have to be set up.

Note: For proper operation the gain low and high Limits for individual SSPA have to be set

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- Transmit parameters.
- --- Meter readings.
- --- Protection switch parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not

be changed under normal circumstances.

- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- minGain --- Defines the minimum available gain for the connected PAs.
- maxGain --- Defines the maximum available gain for the connected PAs.
- redundancy --- Defines the redundancy type used (1:1 or 2:1).
- bucNumber --- Defines which specific BUCs the device controls (BUC-1,BUC-2 or BUC-3).
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.bucVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.minGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.maxGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.redundancy	CHOICE	SAVE SETUP	EnumRange (1:1 2:1)
config.bucNumber	CHOICE	SAVE SETUP	EnumRange (BUC-1 BUC-2 BUC-3)
config.deviceComment	TEXT	R/O	StringRange R/O
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange (MANUAL AUTO)
location	CHOICE		EnumRange (STANDBY ONLINE)
position	CHOICE	R/O	EnumRange R/O (A B)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O DIS (A B)
reset	TEXT	NOPRESET	StringRange
internal.forwardPower	TEXT	R/O	StringRange R/O
internal.on	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.deltaGain	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.attn	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Out of lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over Power)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Latched 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Latched 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched 3)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.361 Miteq-DVM100

This page describes the device driver and the device window for the Miteq DVM-100 Digital Video Broadcast Modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DVM-100. (via front panel menu: [Setup] [RMT/LCL] [Remote Port])

parameter	***description***
address	The modulator's packet address. Valid addresses are 65..95 decimal. The device address is assigned via the front panel. Enter the appropriate ASCII character and not the number in this field (e.g. A for 65)
Serial	Remember to set the line parameters for the serial interface connected to the DVM-100 regarding to the values set on the front panel, e.g. *9600,N,8,1* .

Communication protocol / Interface type

The DVM100 device driver requires the *Miteq-MOD95-Protocol* . The DVM100 uses a RS232 ot RS485 interface for M&C.

Installation

Before the sat-nms software will recognize the modulator, you have to set the communication parameters at the unit's front panel. Set the device address to the chosen value and configure the parameters of the serial line. It is not necessary to put the DVM-100 into an "remote" mode, because this is done automatically if the modulator receives the first remote command. You can verify this by control the LED's labeled RMT (remote mode) and LCL (local mode) on the front panel.

Remote command list and implementation status

3-1 : remote/local communication commands

- cks --- config.checksum --- checksum [cks]
- dis --- config.display --- remote display [dis]
- llo --- config.lockout --- set local lockout
- pwd --- config.password --- set password

3-2 : system setup commands

- exe --- [used with set] ---execute
- rcl --- [not inpl.] --- recall setup
- rst --- reset --- reset
- sav --- [not impl.] --- save setup
- set:adapter --- config.rate adapter --- set rate adapter
- set:alpha --- config.alpha --- set roll of factor
- set:bitrate --- tx.mod.dataRate --- set input data bit rate
- set:buffersize --- config.fifosize --- set average fifo level
- set:if:freq --- tx.frequency --- set if frequency
- set:if:level --- tx.power --- set if level

- set:if:mode --- tx.on --- set if operation mode
- set:if:spectrum --- tx.mod.spectrumInvert --- set if spectrum polarity
- set:inp:mode --- tx.ifc.framingMode ---set input framing mode
- set:inp:port --- tx.ifc.hardware --- select input data port
- set:modtype --- tx.mod.type --- set modulation type
- set:output.port --- setup.outputport --- select input to be routed out
- set:ref --- refClkSrc --- select 10 mhz reference
- set:symbolrate --- tx.mod.symbolrate --- set output symbol rate
- set:syncbyte --- [not impl.] --- set sync byte value

3-3 i/o switchover commands

- set:if:swmode --- [not impl.] --- set if output switchover mode
- set:if:fbmode --- [not impl.] --- set if output fallback mode
- set:if:fbtime --- [not impl.] --- set if output fallback time
- set:inp:swmode --- [not impl.] --- set asi input switchover
- set:inp:fbmode --- [not impl.] --- set asi input fallback mode
- set:inp:fbtime --- [not impl.] --- set asi input fallback time
- set:output:swmode --- [not impl.] --- set asi output switchover
- set:output:fbmode --- [not impl.] --- set asi output fallback mode
- set:output:fbtime --- [not impl.] --- set asi output fallback time

3-4 status commands

- alm --- [internal.state] --- alarm query
- cer --- [not impl.] --- command error buffer query
- cfg:alm:* --- [not impl.] --- configure alarm processing
- cfg:alm:data --- [not impl.] --- fault handling of data alarm
- cfg:alm:fifo --- [not impl.] --- fault handling of fifo alarm
- cfg:alm:sync --- [not impl.] --- fault handling of sync alarm
- cfg:alm:lvl --- [not impl.] --- fault handling of IF level alarm
- cfg:alm:lock --- [not impl.] --- fault handling of lock alarm
- cfg:alm:ref --- [not impl.] --- fault handling of reference alarm
- cfg:alm:comm --- [not impl.] --- fault handling of fifo alarm
- cfg:s_n --- [not impl.] --- configure s/n factory setting
- cls --- [not impl.] --- clear status
- idn --- info.vendor --- identification query
- idn --- info.firmware --- identification query
- idn --- info.serial --- identification query
- larm --- [not impl.] --- latched alarm query
- mon:dcps --- info.powersupply --- query dc power supply query
- mon:dly --- info.packetdelay --- peak delay query
- mon:fifo --- info.usedfifo --- fifo space used query
- mon:port --- info.activeports --- query active port
- mon:rate --- info.inputrate --- measured input rate query
- mon:swt --- [not impl.] --- query if switch position
- oer --- [not impl.] --- operational error query
- rmt --- [not impl.] --- remote query

- tst --- [not impl.] --- test query

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.checksum	CHOICE	SETUP	EnumRange (ON OFF)
config.display	CHOICE	SETUP	EnumRange (ON OFF)
config.lockout	CHOICE	SETUP	EnumRange (ON OFF)
config.password	INTEGER	SETUP	IntegerRange (0 .. 99999)
config.rateadapter	CHOICE	SETUP	EnumRange (ON OFF)
config.alpha	FLOAT	SETUP	DoubleRange (0.15 .. 0.35)
config.fifosize	INTEGER	SETUP	IntegerRange (2 .. 4000)
config.outputport	CHOICE	SETUP	EnumRange (A-EQ A-DIR B-EQ B-DIR)
reset	TEXT	NOPRESET	StringRange
info.vendor	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.powersupply	TEXT	R/O	StringRange R/O
info.packetdelay	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.usedfifo	TEXT	R/O	StringRange R/O
info.activeports	TEXT	R/O	StringRange R/O
info.inputrate	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
modemType	CHOICE	R/O	EnumRange R/O (DVB)
refClkSrc	CHOICE		EnumRange (INT EXT)
refClkFreq	FLOAT	R/O	DoubleRange R/O (1.000000 .. 10.000000)
tx.mod.dataRate	FLOAT		DoubleRange (0.916667 .. 108.500000)
tx.mod.symbolRate	FLOAT		DoubleRange (1.000000 .. 31.000000)
tx.mod.type	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-7/8 8PSK-2/3 8PSK-5/6 8PSK-8/9 16QAM-3/4 16QAM-7/8)
tx.mod.modulation	CHOICE		EnumRange (QPSK 8PSK 16QAM)
tx.mod.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8 8/9)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.ifc.framingMode	CHOICE		EnumRange (187 188 204)
tx.ifc.hardware	CHOICE		EnumRange (ASI-A ASI-B ASI-O RS422 SPI)
tx.frequency	FLOAT		DoubleRange (50.2000 .. 88.0000)
tx.power	FLOAT		DoubleRange (-20.0 .. 0.0)
tx.on	CHOICE		EnumRange (MOD CW OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Data)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fifo)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sync)
faults.04	ALARM	R/O	AlarmFlagRange R/O (IF level)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Synth. out of lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (External reference)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Communication lost)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Built in test)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.type	TEXT	R/O	StringRange R/O

9.362 Miteq-Downconverter-9900-SNMP

This page describes the device driver and the device window for the Miteq down-converters 9900 series via SNMP It detects the frequency range and sets it´s range accordingly.

It was tested with a C-Band downconverter:

- SW revision 172138 v1.057

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- gainRange --- Adjust the gain range to match the respective downconverter.
- ifFrequency --- Sets if frequency between 70 MHz and 140 MHz.
- impedance --- Sets units impedance between 50 Ohm and 75 Ohm.
- band --- Sets the band number between 1 up to 16
- frequencyDac--- Sets external frequency reference DAC .
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.name	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.xilinx	TEXT	R/O	StringRange R/O
info.unitId	TEXT	R/O	StringRange R/O
info.bands	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.loFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lowFreqEdge	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.highFreqEdge	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifFrequency	CHOICE	SETUP	EnumRange (70 MHz 140 MHz)
config.impedance	CHOICE	SETUP	EnumRange (50 Ohm 75 Ohm)
config.band	INTEGER	SETUP	IntegerRange (1 .. 16)
config.frequencyDac	INTEGER	SETUP	IntegerRange (0 .. 4095)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (ON OFF)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.slope	FLOAT		DoubleRange (0.0 .. 30.0)
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Test)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Logged)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LO Level)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amp Current)
faults.07	ALARM	R/O	AlarmFlagRange R/O (External)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module Comms)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.363 Miteq-Downconverter-9900

This page describes the device driver and the device window for the Miteq down-converters 9900 series via Miteq-MOD95 e.g.:

- D9901-1-1K,
- D9908-6-1K.

It was tested with Ku Band version with SW:**Rev 1.121**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
ifFrequency	Sets if frequency between 70 MHz and 140 MHz.
gainRange	Adjust the gain range to match the respective downconverter.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.unitId	TEXT	R/O	StringRange R/O
info.reference	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
info.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltage.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltage.4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.ifFrequency	CHOICE	SETUP	EnumRange (70 MHz 140 MHz)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (ON OFF)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Logged)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU)
faults.06	ALARM	R/O	AlarmFlagRange R/O (LO Level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amp Current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (External)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module Comms)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.364 Miteq-KBand-Upconverter

This page describes the device driver and the device window for the Miteq K-Band Up-converters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel as ASCII-character (e.g. @ for 64dec).
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be bridged to RS232 inside.

Installation

Before you can operate the up-converter with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (17300.000 .. 18100.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)

9.365 Miteq-KuBand-Upconverter

This page describes the device driver and the device window for the Miteq Ku-Band Up-converters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel as ASCII-character (e.g. @ for 64dec).
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be bridged to RS232 inside.

Installation

Before you can operate the up-converter with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (13750.000 .. 14500.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.366 Miteq-LBand-Downconverter

This page describes the device driver and the device window for the Miteq l-band down-converter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (950.000 .. 2150.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)

9.367 Miteq-LBand-Upconverter

This page describes the device driver and the device window for the Miteq l-band up-converters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (950.000 .. 2150.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.368 Miteq-NSU

This page describes an addon device driver and the device window for the Miteq NSU* upconverter redundancy controller.

The driver works with the 1:1, 2:1 and N:1 variants of the device. It automatically recognizes the number of chains controlled by the device and grays out the parameters of all unused chains. This is done once after the driver establishes communication to the device.

Beside the redundancy switch control the driver controls a number of settings for each upconverter connected to the Miteq NSU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the redundancy switching parameters and the parameters of the connected upconverters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.

- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.devicename	TEXT	R/O	StringRange R/O
info.switchtype	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
internal.stby	TEXT	R/O	StringRange R/O
testSlope	TEXT	R/O	StringRange R/O
avail.R	TEXT	R/O	StringRange R/O
avail.1	TEXT	R/O	StringRange R/O
avail.2	TEXT	R/O	StringRange R/O
avail.3	TEXT	R/O	StringRange R/O
avail.4	TEXT	R/O	StringRange R/O
avail.5	TEXT	R/O	StringRange R/O
avail.6	TEXT	R/O	StringRange R/O
avail.7	TEXT	R/O	StringRange R/O
avail.8	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
avail.9	TEXT	R/O	StringRange R/O
avail.10	TEXT	R/O	StringRange R/O
avail.11	TEXT	R/O	StringRange R/O
avail.12	TEXT	R/O	StringRange R/O
serialLinkFault.R	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.1	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.2	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.3	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.4	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.5	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.6	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.7	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.8	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.9	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.10	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.11	BOOLEAN	R/O	BooleanRange R/O (true false)
serialLinkFault.12	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.R	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.1	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.2	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.3	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.4	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.5	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.6	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.7	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.8	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.9	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
converterFault.10	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.11	BOOLEAN	R/O	BooleanRange R/O (true false)
converterFault.12	BOOLEAN	R/O	BooleanRange R/O (true false)
position.R	TEXT	NOPRESET	StringRange
position.1	TEXT	NOPRESET	StringRange
position.2	TEXT	NOPRESET	StringRange
position.3	TEXT	NOPRESET	StringRange
position.4	TEXT	NOPRESET	StringRange
position.5	TEXT	NOPRESET	StringRange
position.6	TEXT	NOPRESET	StringRange
position.7	TEXT	NOPRESET	StringRange
position.8	TEXT	NOPRESET	StringRange
position.9	TEXT	NOPRESET	StringRange
position.10	TEXT	NOPRESET	StringRange
position.11	TEXT	NOPRESET	StringRange
position.12	TEXT	NOPRESET	StringRange
switch.1	CHOICE	R/O	EnumRange R/O (A B)
switch.2	CHOICE	R/O	EnumRange R/O (A B)
switch.3	CHOICE	R/O	EnumRange R/O (A B)
switch.4	CHOICE	R/O	EnumRange R/O (A B)
switch.5	CHOICE	R/O	EnumRange R/O (A B)
switch.6	CHOICE	R/O	EnumRange R/O (A B)
switch.7	CHOICE	R/O	EnumRange R/O (A B)
switch.8	CHOICE	R/O	EnumRange R/O (A B)
switch.9	CHOICE	R/O	EnumRange R/O (A B)
switch.10	CHOICE	R/O	EnumRange R/O (A B)

name	*type*	*flags*	*range*
switch.11	CHOICE	R/O	EnumRange R/O (A B)
switch.12	CHOICE	R/O	EnumRange R/O (A B)
priority.R	INTEGER		IntegerRange (0 .. 0)
priority.1	INTEGER		IntegerRange (0 .. 0)
priority.2	INTEGER		IntegerRange (0 .. 0)
priority.3	INTEGER		IntegerRange (0 .. 0)
priority.4	INTEGER		IntegerRange (0 .. 0)
priority.5	INTEGER		IntegerRange (0 .. 0)
priority.6	INTEGER		IntegerRange (0 .. 0)
priority.7	INTEGER		IntegerRange (0 .. 0)
priority.8	INTEGER		IntegerRange (0 .. 0)
priority.9	INTEGER		IntegerRange (0 .. 0)
priority.10	INTEGER		IntegerRange (0 .. 0)
priority.11	INTEGER		IntegerRange (0 .. 0)
priority.12	INTEGER		IntegerRange (0 .. 0)
frequency.R	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.1	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.2	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.3	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.4	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.5	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.6	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.7	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.8	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.9	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
frequency.10	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.11	FLOAT		DoubleRange (0.000 .. 0.000)
frequency.12	FLOAT		DoubleRange (0.000 .. 0.000)
eq.R	TEXT	R/O	StringRange R/O
eq.1	FLOAT		DoubleRange (0.0 .. 0.0)
eq.2	FLOAT		DoubleRange (0.0 .. 0.0)
eq.3	FLOAT		DoubleRange (0.0 .. 0.0)
eq.4	FLOAT		DoubleRange (0.0 .. 0.0)
eq.5	FLOAT		DoubleRange (0.0 .. 0.0)
eq.6	FLOAT		DoubleRange (0.0 .. 0.0)
eq.7	FLOAT		DoubleRange (0.0 .. 0.0)
eq.8	FLOAT		DoubleRange (0.0 .. 0.0)
eq.9	FLOAT		DoubleRange (0.0 .. 0.0)
eq.10	FLOAT		DoubleRange (0.0 .. 0.0)
eq.11	FLOAT		DoubleRange (0.0 .. 0.0)
eq.12	FLOAT		DoubleRange (0.0 .. 0.0)
slope.R	TEXT	R/O	StringRange R/O
slope.1	FLOAT		DoubleRange (0.0 .. 0.0)
slope.2	FLOAT		DoubleRange (0.0 .. 0.0)
slope.3	FLOAT		DoubleRange (0.0 .. 0.0)
slope.4	FLOAT		DoubleRange (0.0 .. 0.0)
slope.5	FLOAT		DoubleRange (0.0 .. 0.0)
slope.6	FLOAT		DoubleRange (0.0 .. 0.0)
slope.7	FLOAT		DoubleRange (0.0 .. 0.0)
slope.8	FLOAT		DoubleRange (0.0 .. 0.0)
slope.9	FLOAT		DoubleRange (0.0 .. 0.0)
slope.10	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
slope.11	FLOAT		DoubleRange (0.0 .. 0.0)
slope.12	FLOAT		DoubleRange (0.0 .. 0.0)
atten.R	FLOAT		DoubleRange (0.0 .. 30.0)
atten.1	FLOAT		DoubleRange (0.0 .. 30.0)
atten.2	FLOAT		DoubleRange (0.0 .. 30.0)
atten.3	FLOAT		DoubleRange (0.0 .. 30.0)
atten.4	FLOAT		DoubleRange (0.0 .. 30.0)
atten.5	FLOAT		DoubleRange (0.0 .. 30.0)
atten.6	FLOAT		DoubleRange (0.0 .. 30.0)
atten.7	FLOAT		DoubleRange (0.0 .. 30.0)
atten.8	FLOAT		DoubleRange (0.0 .. 30.0)
atten.9	FLOAT		DoubleRange (0.0 .. 30.0)
atten.10	FLOAT		DoubleRange (0.0 .. 30.0)
atten.11	FLOAT		DoubleRange (0.0 .. 30.0)
atten.12	FLOAT		DoubleRange (0.0 .. 30.0)
mute.R	CHOICE		EnumRange (MUTED NOT MUTED)
mute.1	CHOICE		EnumRange (MUTED NOT MUTED)
mute.2	CHOICE		EnumRange (MUTED NOT MUTED)
mute.3	CHOICE		EnumRange (MUTED NOT MUTED)
mute.4	CHOICE		EnumRange (MUTED NOT MUTED)
mute.5	CHOICE		EnumRange (MUTED NOT MUTED)
mute.6	CHOICE		EnumRange (MUTED NOT MUTED)
mute.7	CHOICE		EnumRange (MUTED NOT MUTED)
mute.8	CHOICE		EnumRange (MUTED NOT MUTED)
mute.9	CHOICE		EnumRange (MUTED NOT MUTED)
mute.10	CHOICE		EnumRange (MUTED NOT MUTED)

name	*type*	*flags*	*range*
mute.11	CHOICE		EnumRange (MUTED NOT MUTED)
mute.12	CHOICE		EnumRange (MUTED NOT MUTED)
protection	CHOICE		EnumRange (MANUAL AUTO)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Converter bus)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Converter contact)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch module)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch bus)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote control)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Converter R)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Converter 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Converter 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Converter 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Converter 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Converter 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Converter 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Converter 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Converter 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Converter 9)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Converter 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Converter 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Converter 12)

9.369 Miteq-TLT-LO

This page describes the device driver and the device window for the Miteq Test Loop Translator DN9400 series via Miteq-MOD95.

It was tested with:

- DN-14125-11450-C22442

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- The meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
lo.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.on	CHOICE		EnumRange (ON OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.02	ALARM	R/O	AlarmFlagRange R/O (LO A)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO B)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (IF LO Level Detect)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF LO Level Detect)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.370 Miteq-TLT

This page describes the device driver and the device window for the Miteq Test Loop Translator with synthesizeable LO frequency DN9700 series via Miteq-MOD95.

It was tested with:

- DN-97 17.7/12.1-C23691

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- The meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.ifFrequency	CHOICE	R/O	EnumRange R/O (70MHz 140MHz)
info.voltageP.15.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltageA.5.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltageB.5.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltageN.15.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.extRef	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	INTEGER		IntegerRange (0 .. 30)
tx.attn	INTEGER		IntegerRange (0 .. 30)
lo.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Logged alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Redundancy Bus)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Redundancy switch)
faults.08	ALARM	R/O	AlarmFlagRange R/O (External)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX OFF)

9.371 Miteq-UPC

This page describes the device driver and the device window for the Miteq Uplink Power Control Unit.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
channels	number of enabled channels
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.channels	INTEGER	SAVE SETUP	IntegerRange (1 .. 10)
algorithm	CHOICE		EnumRange (OPEN-LOOP CLOSED-LOOP COMPARISION DUAL-TRACK DIVERSITY-SWITCHING)
sampleTime	FLOAT		DoubleRange (0.2 .. 10.0)
input.a	CHOICE		EnumRange (OFF STANDBY ACTIVE)
input.b	CHOICE		EnumRange (OFF STANDBY ACTIVE)
inputPower.a	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
inputPower.b	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
activeInput	CHOICE		EnumRange (NONE A B A+B)
channel.01	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.02	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.03	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.04	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.05	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.06	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.07	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.08	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.09	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
channel.10	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
clearSkyAttn.01	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.02	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.03	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.04	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.05	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.06	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.07	FLOAT		DoubleRange DIS (0.0 .. 20.0)

name	*type*	*flags*	*range*
clearSkyAttn.08	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.09	FLOAT		DoubleRange DIS (0.0 .. 20.0)
clearSkyAttn.10	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.01	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.02	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.03	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.04	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.05	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.06	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.07	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.08	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.09	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.10	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.01	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.02	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.03	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.04	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.05	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.06	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.07	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.08	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.09	FLOAT		DoubleRange DIS (0.0 .. 20.0)
maxStepSize.10	FLOAT		DoubleRange DIS (0.0 .. 20.0)
upcMax.01	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.02	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.03	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
upcMax.04	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.05	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.06	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.07	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.08	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.09	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upcMax.10	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
internal.inA	TEXT	R/O	StringRange R/O
internal.inB	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel 01)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Channel 02)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel 03)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Channel 04)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Channel 05)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Channel 06)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Channel 07)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Channel 08)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Channel 09)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Channel 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Receiver A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Receiver B)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power supply A)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power supply B)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.372 Miteq-Upconverter-9900-SNMP

This page describes the device driver and the device window for the Miteq up-converters 9900 series via SNMP

It detects the frequency range and sets it's range accordingly.

It was tested with a C-Band upconverter

- SW revision 172138 v1.135

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- gainRange --- Adjust the gain range to match the respective downconverter.

- ifFrequency --- Sets if frequency between 70 MHz and 140 MHz.
- impedance --- Sets units impedance between 50 Ohm and 75 Ohm.
- band --- Sets the band number between 1 up to 16
- frequencyDac--- Sets external frequency reference DAC .
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.xilinx	TEXT	R/O	StringRange R/O
info.unitId	TEXT	R/O	StringRange R/O
info.bands	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.loFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lowFreqEdge	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.highFreqEdge	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
info.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltage.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.voltage.4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifFrequency	CHOICE	SETUP	EnumRange (70 MHz 140 MHz)
config.impedance	CHOICE	SETUP	EnumRange (50 Ohm 75 Ohm)
config.band	INTEGER	SETUP	IntegerRange (1 .. 16)
config.frequencyDac	INTEGER	SETUP	IntegerRange (0 .. 4095)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (ON OFF)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.slope	FLOAT		DoubleRange (0.0 .. 30.0)
internal.gain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Logged)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LO Level)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amp Current)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (External)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module Comms)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.373 Modcod-Splitter

This page describes the device driver and the device window for the*Modcod-Splitter* logical device. This device splits up a combined MODCOD setting into individual parameters

Modulator / demodulator devices offer parameters to select the modulation type (e.g. QPSK), the FEC setting (e.g. 3/4) and newer ones also the modulation standard aka. DVB-mode setting (e.g. DVBS2). Many device use individual parameter for this but there are also device types which combine these parameters to one MODCOD setting.

The *Modcod-Splitter* device may be used to split up the MODCOD setting into individual parameters. It supports MODCOD parameters like QPSK-3/4 (modulation-type followed by FEC) and like DVBS-QPSK-3/4 (DVB-mode followed by modulation-type followed by FEC). The device automatically reads and parsed the choices offered by the target parameter and sets the choices for it's individual parameters accordingly.

Best match translation

When the operator changes one of the parameters DVBMode, modulation or FEC, this may lead to a MODCOD combination the target device does not support. the sat-nms standard behaviour for such a case would be to fall back to the very first MODCOD setting of the device. This certainly is not what an operator expects when he changes parameters.

Hence, the *Modcod-Splitter* device implements a 'best match' translation algorithm which changes invalid MODCOD settings to a valid setting which resembles as much as possible from the combination the operator selected. This is done from left to right, so in many cases only the FEC rate is changed to a valid setting.

Because of the left to right priority of this algorithm, the best way to operate the*Modcod-Splitter* device manually is to set the DVB-mode first, then the modulation-type and finally the FEC.

Delayed command forwarding

In many cases the *Modcod-Splitter* device will be used allow the automated control of a MODCOD parameter from another logical device which expects separate parameters instead of a MODCOD setting. Such a device will send the parameters at the same time to the *Modcod-*

Splitter , but they may arrive there in random order.

For this reason the *Modcod-Splitter* device does not update the target parameter immediately with each parameter change it recognizes. It waits at least 750 msec (the exact time depends on the IDLE time setting of the thread/interfaces where the device is configured) after a parameter arrived before the MODCOD setting gets combined with the 'best match' algorithm and sent to the target device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the individual DVB-mode, modulation-type and FEC parameters
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there is exactly one configuration parameters which must be set to integrate the device into the VLC/M&C:

parameter	***description***
modcodId	The parameter ID of the target parameter this device shall control.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.modcodId	TEXT	SAVE SETUP	StringRange
dvbMode	CHOICE		EnumRange (UNKNOWN)
modulation	CHOICE		EnumRange (UNKNOWN)
fec	CHOICE		EnumRange (UNKNOWN)

9.374 Modem-Protection-Controller

The Modem-Proection-Controller logical device builds together with the Protection-Switch-N-To-M device a n:m redundancy solution for satellite modulators, which supports a pool of backup modems from which one modem is taken in case of failure of an operative modem. In case of a redundancy switch all modem parameters are mirrored to the spare modem, so all modems in the pool (active and spare ones) must be exactly of the same type.

For n:m redundancy, one Protection-Switch-N-To-M device and for each managed modulator a Modem-Protection-Controller device are configured in the M&C device setup. This distribution across two device types enables a very flexible configuration of the redundancy, so there is no explicit limit on the number of managed modulators.

The devices are based on the following device configuration:

- The user has a freely definable number of identical modulators, some of which act as spare devices, the rest are operational.
- The modulators receive their input signal via IP, so that when switching, the input signal is also switched by mirroring the modulator parameters.
- The outputs of all modulators are wired to the inputs of a crosspoint-based switching matrix.

In the event of a switchover, all crosspoints belonging to the first modulator are copied to the second one, and all crosspoints of the first modulator are deactivated.

The Modem-Protection-Controller device carries out the actual task of monitoring and redundancy switching. One Modem-Protection-Controller device is configured for each of the modulators involved. Whether the individual modulator is an operational or a spare device can be switched "on the fly" using a configuration parameter. In addition, a modem protection controller device can be configured to the INACTIVE state, which designates a device that is neither PRIMARY nor SPARE, e.g. as a placeholder for later extensions or if a modulator is temporarily unavailable.

Each Modem-Protection-Controller device monitors two configurable fault flags. The first faul is labeled "general fault" and should combine all fault which prevent a modem from being used. The second fault ("operational fault") should be active with fault conditions that shall trigger a redundancy switching but shall not prevent a spare modem from being used if this fault is active. A missing input signal is an example for this type of fault.

When performing a redundancy switch, the device first requests a free spare modulator from the Protection-Switch-N-to-M device. It mirrors the settings of the modulator to be replaced on this and then copies the associated matrix crosspoints in such a way that the output of the spare modulator is routed to the destinations that were previously served by the replaced modulator. The output of the replaced device is separated from all destinations in the matrix.

The redundancy switch can also be triggered manually by clicking REPLACE-BY-SPARE. In the same way, after the replaced device has been repaired/checked, it can be switched back to operational mode (SWITCH-BACK). The original matrix configuration of the device is restored, but no modulator parameters from the spare device are mirrored back to the primary one.

Modem controllers configured as SPARE may be bound to a certain primary modem. In this

case the spare modem is no longer part of the spare modem pool, instead it acts as a 1:1 backup for the selected primary modem. Use the 'reserved for' parameter to configure this feature. Modem controllers which have the 'reserved for' parameter set to NONE are part of the spare modem pool.

By default modem parameters are mirrored after the primary modem has been set offline at the switch matrix and before the backup modem gets online. For 1:1 modem pairs a different approac may be configured. If the '121MirrorMode' config parameter is set to IMMEDIATE at the primary modem, mirroring is skipped during the switchover procedure, thus speeding this up significantly. Instead mirroring the modem parameter is triggered by the following events:

- A backup modem is configured as 'reserved for' for this primary modem
- The primary modem changes its 'tx.on' parameter to ON (this assumes, that configuring this device has been completed).
- The operator clicks the MIRROR NOW button.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. * --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
protectionSwitch	The name of the Protection-Switch-N-To-M device, this Modem-Protection-Controller is assigned to.
modem	The name of the modem / modulator device this Modem-Protection-Controller monitors anf controls.
role	The role of this modulator. One of PRIMARY, SPARE or INACTIVE.
matrixPort	The number of the matrix input the modulator output is connected to. The number must be writen with a leading zero for port numbers 01..09, the device expects the port number as it appears in the variable name of the matrix device driver.

parameter	*description*
faultId.01	The parameter ID to monitor for the "general fault". This fault is monitored by both, PRIMARY and SPARE modems. If a fault is pending, it triggers a redundancy switch or prevents a modem being taken from the pool of spare modems, depending on the modem's role.
faultId.02	The parameter ID to monitor for the "operational fault". This fault is recognized by PRIMARY modems, it also trigers an redundancy switch wif pending. The "operational fault" id ignored by SPARE modems.
preMirrorDelay	The time (msecs) between switching OFF all matrix routes originating from the SPARE mode before mirroring the parameters of the PRIMARY modem to the SPARE one. This delay shall ensure that the SPARE modem is not on air when it gets configured for its next use.
postMirrorDelay	The time (msecs) between mirroring the parameters of the PRIMARY modem to the SPARE one and mirroring the matrix port settings. This delay shall ensure that the SPARE modem is completly configured before it gets switched on air.
121MirrorMode	The way (SWITCHOVER, IMMEDIATE) the device handles the mirroring of modem parameters in a 1:1 redundancy configuration

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.protectionSwitch	TEXT	SAVE SETUP	StringRange
config.modem	TEXT	SAVE SETUP	StringRange
config.role	CHOICE	SAVE SETUP	EnumRange (INACTIVE PRIMARY SPARE)
config.matrixPort	TEXT	SAVE SETUP	StringRange
config.faultId.01	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.faultId.02	TEXT	SAVE SETUP	StringRange
config.preMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.postMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.121MirrorMode	CHOICE	SAVE SETUP	EnumRange (SWITCHOVER IMMEDIATE)
protection	CHOICE	SAVE	EnumRange (OFF ON)
replaceBySpare	TEXT	NOPRESET	StringRange
switchBack	TEXT	NOPRESET	StringRange
forceMirror	TEXT	NOPRESET	StringRange
reservedFor	CHOICE		EnumRange DIS (NONE)
comment	TEXT	SAVE	StringRange
reservedFor.save	TEXT	SAVE	StringRange
modemFault	BOOLEAN	R/O	BooleanRange R/O (true false)
externalFault	BOOLEAN	R/O	BooleanRange R/O (true false)
inUse	BOOLEAN	R/O	BooleanRange R/O (true false)
state	TEXT	R/O	StringRange R/O
mayForceMirror	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Modem general)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Modem operational)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.375 Monitor-Switch

This page describes the device driver and the device window for the*Monitor-Switch* logical device. This device can be used to setup a monitoring device (e.g. a test converter or a measuring device) to measure the signal from a certain signal chain in the station. The *Monitor-*

Switch logical device controls a N:1 signal switch in front of the monitoring device in order to select the appropriate signal and it copies one parameter value (typically a frequency) from an arbitrary device in each chain to the testing / monitoring device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets / shows the monitor switch position.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
choices	The list of choices presented to the operator for the 'position' parameter. For each signal to monitor, a unique name must be provided. The list entries must be separated by comma characters. The device will automatically add an "UNKNOWN" selection in front of the given list of entries. Example: " **PROG-1,PROG-2,PROG-3** "
sourceIds	A comma separated list of parameter identifiers. The device will use the parameter id at the same list position as the selected 'position' value owns in the choices list above and will copy the corresponding parameter value to the destination parameter defined below. Example: " **UPC-1.tx.frequency,UPC-2.tx.frequency,UPC-3.tx.frequency** "
destinationId	The ID of the parameter to copy the source value to. Example: "**TEST-CONVERTER.frequency** "
positions	A comma separated list of switch positions which are applied to the input signal selection switch in front of the monitoring device. Example: " **1,2,3** "
switchId	The parameter ID controlling the position of the switch in front of the monitoring device. Example: " **S05.position** "

Remarks

- The *Monitor-Switch* device does not read back the switch position. This is because the 'positions' translation table is not necessarily one-to-one, the primary position cannot be derived non-ambiguously from the hardware switch position. As a consequence, the

Monitor-Switch device wakes up after system boot showing the position "UNKNOWN".
- The destination parameter is only set when the position parameter is commanded. Later changes of the source parameter assigned to the selected position are ignored.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.choices	TEXT	SAVE SETUP	StringRange
config.sourceIds	TEXT	SAVE SETUP	StringRange
config.destinationId	TEXT	SAVE SETUP	StringRange
config.positions	TEXT	SAVE SETUP	StringRange
config.switchId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (UNKNOWN)

9.376 Monterey-140

This page describes the device driver and the device window for the Chaparral Monterey-140 receiver / antenna controller. The sat-nms driver for this device only sets the position of a polarmount antenna which is controlled by the Monterey-140. The driver does not control any other parameter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page controls the antenna pointing. Please note that the displayed pointing is the recently commanded value. The Monterey-140 does not provide the possibility to read back the actual antenna position via the RS232 interface. After power on the the position field is empty for this reason.The Monterey-140 accepts integer position values from 0 to 9999. The position value at the parameter page has an offset included which is set at the configuration page. Typical position values with the offset included are -400 .. +400.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
loLimit	The lowest position value the controller accepts.
hiLimit	The highest position value the controller accepts.
reference	The reference / offset value. If this value is set to 4500 (the default) and a position of 200 is commanded to the device driver, the driver sets 4700 at the device.
Serial	Remember to set the line parameters for the serial interface connected to the controller. The Monterey-140 expects *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.loLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 9999)
config.hiLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 9999)
config.reference	INTEGER	SAVE SETUP	IntegerRange (0 .. 9999)
internal.p	INTEGER	R/O	IntegerRange R/O (0 .. 0)
position	INTEGER		IntegerRange (-9999 .. 9999)

9.377 Moxa-ioLogik-1200

This page describes the device driver and the device window for the*Moxa ioLogik E12xx* data acquisition module family. The device driver monitors and controls the digital inputs/outputs of various ioLogik E12xx modules. Analog or temperature sensor input functions are not covered by this driver, neither are counter/timer timer function of the some modules. The driver can be used together with the following modules:

- E1210 (16 inputs)
- E1212 (8 inputs, 8 outputs)
- E1214 (6 inputs, 6 relay outputs)

The *Moxa-ioLogik-1200* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Moxa-ioLogik-1200* device driver directly (except to access the temperature readings). There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *SatService-IO-FEP* device driver as a low level driver to access the physical I/O ports.

Input ports

The variable I00001 refers to the input ports provided by the device. The least significant bit denotes the state of DI0.

The variable reports the physical state if the input. A bit is set to '1' if the input is open or externally driven to high level. The bit reads '0' if the input is connected to GND or externally driven to low level.

Output ports

The variable 'Q00001' reports the actual state of the digital outputs and also sets the output states. The least significant bit in the variable corresponds to output DO0.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Moxa ioLogik module acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the device in the device setup window , there must not be defined any other devices beside this module for this interface.
- The Moxa ioLogik modules are controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1
moduleType	The actual module type must be selected to tell the driver how many inputs/outputs are available with this module.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.moduleType	CHOICE	SAVE SETUP	EnumRange (E1210 E1212 E1214 E1260)
config.sensors	CHOICE	SAVE SETUP	EnumRange (0 1 2 3 4 5 6)
I00001	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
Q00001	HEX		HexRange DIS (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
temp.1	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temp.2	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temp.3	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temp.4	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temp.5	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
temp.6	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.1	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.2	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.3	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.4	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.5	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
minTemp.6	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.1	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.2	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.3	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.4	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.5	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
maxTemp.6	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
i	HEX	R/O	HexRange R/O (0 .. 0)

9.378 Moxa-ioLogik-2214

This page describes the device driver and the device window for the*Moxa ioLogic E2214* I/O control unit. The device driver monitors and controls the digital inputs/outputs of the unit. Pulse/PWM output functions are not covered by this driver, neither are counter/timer timer functions.

The *Moxa-ioLogik-2214* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the *Moxa-ioLogik-2214* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Moxa-ioLogik-2214* device driver as a low level driver to access the physical I/O ports.

Input ports

The Moxa ioLogik 2214 device provides 6 digital inputs. The driver shows the state of these inputs in the bits of the variable I00001. The variable reports the physical state if the inputs. Bit 0 (LSB) corresponds to input 1. A bit is set to '1' if the input is open or externally driven to high level. The bit reads '0' if the input is connected to GND or externally driven to low level.

Output ports

The Moxa ioLogik 2214 device provides 6 digital outputs. The driver shows the state of these outputs in the bits of the variable Q00001. Bit 0 (LSB) corresponds to output 1. Writing to the Q0001 variable with a bit set to '1' means to activate this output, writing a '0' deactivates the output.

Output ports are read back by the driver after the sat-nms software has changed an output bit or every 10 seconds.

Power on configuration

The ioLogik device does not remember the state of an output over a power cycle. Instead, there are 6 configuration parameters controlling the power on state of each individual output. See paragraph 'Configuration parameters' below.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Moxa ioLogik module acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the ioLogik 2214 device in the device setup window , there must not be any other device defined in this interface. This is to ensure the maximum performance and minimal delay in I/O operations.
- The Moxa ioLogik 2214 unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1
poweron.out.1	Power on state of output 1
poweron.out.2	Power on state of output 2
poweron.out.3	Power on state of output 3
poweron.out.4	Power on state of output 4
poweron.out.5	Power on state of output 5
poweron.out.6	Power on state of output 6

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.poweron.out.1	CHOICE	SETUP	EnumRange (OFF ON)
config.poweron.out.2	CHOICE	SETUP	EnumRange (OFF ON)
config.poweron.out.3	CHOICE	SETUP	EnumRange (OFF ON)
config.poweron.out.4	CHOICE	SETUP	EnumRange (OFF ON)
config.poweron.out.5	CHOICE	SETUP	EnumRange (OFF ON)
config.poweron.out.6	CHOICE	SETUP	EnumRange (OFF ON)
I00001	HEX	R/O	HexRange R/O (0 .. 0)
Q00001	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

9.379 Moxa-ioLogik-4000

This page describes the device driver and the device window for the*Moxa ioLogik 4000* data acquisition unit. The device driver monitors and controls the digital inputs/outputs of various modules hosted by the ioLogik 4000 unit. The driver supports digital I/O modules and the M6200 temperature measurement module (see below for details about the latter one).

The *Moxa-ioLogik-4000* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Moxa-ioLogik-4000* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Moxa-ioLogik-4000* device driver as a low level driver to access the physical I/O ports.

Port adressing

The ioLogik device must be configured to work in dynamic address mapping mode to work with the driver. In this mode, each module gets assigned the lowest possible Modbus address. As a result, the modules can be accessed as a continuous sequence of registers.

The driver supports up to 32 16-bit input registers and 32 16-bit output registers appearing as

variables I0000 to I001F and Q0800 to Q081F. The variable names reflect the Modbus addresses of these registers in hexadecimal notation. Beside I0000 to I001F, the driver provides 31 variables called J0000 to J001E containing the input registers shifted by 8 bit. J0000 for example contains the high byte if I0000 as low byte and the low byte of I0001 as high byte.

The Moxa administration tool for the ioLogix 4000 (for the NA-4010 also the Web-GUI of the device) shows the addresses, the device has assigned to each module.

The ioLogix 4000 does not permit to read non-existing registers, hence the device driver needs to know, how many input and output registers the device has configured. Evaluate these numbers from the address map shown in the Moxa administration utility and set the driver configuration variables accordingly.

M-6200 Temperature modules

Beside all digital I/O modules the driver also supports The M-6200 temperature measurement module. However, the device window shows the temperature values in hexadecimal notation (tenths of centigrade), which is hard to read / interpret.

For a temperature display in °C the sat-nms software provides the[Moxa-ioLogik-Temperature] logical device, which translates this value into a readable floating point number. The device also recognizes the special reading '0x8000' which is shown by the module to signal a broken cable to the temperature sensor and generates a fault flag from this.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Moxa ioLogik device acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the ioLogik device in the device setup window, there must not be defined any other devices beside this module for this interface.
- The ioLogik unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msecs for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1
inputRegisters	Must be set to the number of 16-bit input registers, the ioLogik device supports with the actual module configuration. Setting this value too low makes the driver ignore the state of registers (I/O modules) outside this range, setting it too high causes an communication fault because the driver tries to access registers which do not exist in the device.
outputRegisters	Must be set to the number of 16-bit output registers, the ioLogik device supports with the actual module configuration. Setting this value too low makes the driver ignore the state of registers (I/O modules) outside this range, setting it too high causes an communication fault because the driver tries to access registers which do not exist in the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputRegisters	INTEGER	SAVE SETUP	IntegerRange (0 .. 32)
config.outputRegisters	INTEGER	SAVE SETUP	IntegerRange (0 .. 32)
I0000	HEX	R/O	HexRange R/O (0 .. 0)
I0001	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
I0002	HEX	R/O	HexRange R/O (0 .. 0)
I0003	HEX	R/O	HexRange R/O (0 .. 0)
I0004	HEX	R/O	HexRange R/O (0 .. 0)
I0005	HEX	R/O	HexRange R/O (0 .. 0)
I0006	HEX	R/O	HexRange R/O (0 .. 0)
I0007	HEX	R/O	HexRange R/O (0 .. 0)
I0008	HEX	R/O	HexRange R/O (0 .. 0)
I0009	HEX	R/O	HexRange R/O (0 .. 0)
I000A	HEX	R/O	HexRange R/O (0 .. 0)
I000B	HEX	R/O	HexRange R/O (0 .. 0)
I000C	HEX	R/O	HexRange R/O (0 .. 0)
I000D	HEX	R/O	HexRange R/O (0 .. 0)
I000E	HEX	R/O	HexRange R/O (0 .. 0)
I000F	HEX	R/O	HexRange R/O (0 .. 0)
I0010	HEX	R/O	HexRange R/O (0 .. 0)
I0011	HEX	R/O	HexRange R/O (0 .. 0)
I0012	HEX	R/O	HexRange R/O (0 .. 0)
I0013	HEX	R/O	HexRange R/O (0 .. 0)
I0014	HEX	R/O	HexRange R/O (0 .. 0)
I0015	HEX	R/O	HexRange R/O (0 .. 0)
I0016	HEX	R/O	HexRange R/O (0 .. 0)
I0017	HEX	R/O	HexRange R/O (0 .. 0)
I0018	HEX	R/O	HexRange R/O (0 .. 0)
I0019	HEX	R/O	HexRange R/O (0 .. 0)
I001A	HEX	R/O	HexRange R/O (0 .. 0)
I001B	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
I001C	HEX	R/O	HexRange R/O (0 .. 0)
I001D	HEX	R/O	HexRange R/O (0 .. 0)
I001E	HEX	R/O	HexRange R/O (0 .. 0)
I001F	HEX	R/O	HexRange R/O (0 .. 0)
J0000	HEX	R/O	HexRange R/O (0 .. 0)
J0001	HEX	R/O	HexRange R/O (0 .. 0)
J0002	HEX	R/O	HexRange R/O (0 .. 0)
J0003	HEX	R/O	HexRange R/O (0 .. 0)
J0004	HEX	R/O	HexRange R/O (0 .. 0)
J0005	HEX	R/O	HexRange R/O (0 .. 0)
J0006	HEX	R/O	HexRange R/O (0 .. 0)
J0007	HEX	R/O	HexRange R/O (0 .. 0)
J0008	HEX	R/O	HexRange R/O (0 .. 0)
J0009	HEX	R/O	HexRange R/O (0 .. 0)
J000A	HEX	R/O	HexRange R/O (0 .. 0)
J000B	HEX	R/O	HexRange R/O (0 .. 0)
J000C	HEX	R/O	HexRange R/O (0 .. 0)
J000D	HEX	R/O	HexRange R/O (0 .. 0)
J000E	HEX	R/O	HexRange R/O (0 .. 0)
J000F	HEX	R/O	HexRange R/O (0 .. 0)
J0010	HEX	R/O	HexRange R/O (0 .. 0)
J0011	HEX	R/O	HexRange R/O (0 .. 0)
J0012	HEX	R/O	HexRange R/O (0 .. 0)
J0013	HEX	R/O	HexRange R/O (0 .. 0)
J0014	HEX	R/O	HexRange R/O (0 .. 0)
J0015	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
J0016	HEX	R/O	HexRange R/O (0 .. 0)
J0017	HEX	R/O	HexRange R/O (0 .. 0)
J0018	HEX	R/O	HexRange R/O (0 .. 0)
J0019	HEX	R/O	HexRange R/O (0 .. 0)
J001A	HEX	R/O	HexRange R/O (0 .. 0)
J001B	HEX	R/O	HexRange R/O (0 .. 0)
J001C	HEX	R/O	HexRange R/O (0 .. 0)
J001D	HEX	R/O	HexRange R/O (0 .. 0)
J001E	HEX	R/O	HexRange R/O (0 .. 0)
Q0800	HEX		HexRange (0 .. 0)
Q0801	HEX		HexRange (0 .. 0)
Q0802	HEX		HexRange (0 .. 0)
Q0803	HEX		HexRange (0 .. 0)
Q0804	HEX		HexRange (0 .. 0)
Q0805	HEX		HexRange (0 .. 0)
Q0806	HEX		HexRange (0 .. 0)
Q0807	HEX		HexRange (0 .. 0)
Q0808	HEX		HexRange (0 .. 0)
Q0809	HEX		HexRange (0 .. 0)
Q080A	HEX		HexRange (0 .. 0)
Q080B	HEX		HexRange (0 .. 0)
Q080C	HEX		HexRange (0 .. 0)
Q080D	HEX		HexRange (0 .. 0)
Q080E	HEX		HexRange (0 .. 0)
Q080F	HEX		HexRange (0 .. 0)
Q0810	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
Q0811	HEX		HexRange (0 .. 0)
Q0812	HEX		HexRange (0 .. 0)
Q0813	HEX		HexRange (0 .. 0)
Q0814	HEX		HexRange (0 .. 0)
Q0815	HEX		HexRange (0 .. 0)
Q0816	HEX		HexRange (0 .. 0)
Q0817	HEX		HexRange (0 .. 0)
Q0818	HEX		HexRange (0 .. 0)
Q0819	HEX		HexRange (0 .. 0)
Q081A	HEX		HexRange (0 .. 0)
Q081B	HEX		HexRange (0 .. 0)
Q081C	HEX		HexRange (0 .. 0)
Q081D	HEX		HexRange (0 .. 0)
Q081E	HEX		HexRange (0 .. 0)
Q081F	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
i.0	HEX	R/O	HexRange R/O (0 .. 0)
i.1	HEX	R/O	HexRange R/O (0 .. 0)
i.2	HEX	R/O	HexRange R/O (0 .. 0)
i.3	HEX	R/O	HexRange R/O (0 .. 0)
i.4	HEX	R/O	HexRange R/O (0 .. 0)
i.5	HEX	R/O	HexRange R/O (0 .. 0)
i.6	HEX	R/O	HexRange R/O (0 .. 0)
i.7	HEX	R/O	HexRange R/O (0 .. 0)
i.8	HEX	R/O	HexRange R/O (0 .. 0)
i.9	HEX	R/O	HexRange R/O (0 .. 0)
i.a	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
i.b	HEX	R/O	HexRange R/O (0 .. 0)
i.c	HEX	R/O	HexRange R/O (0 .. 0)
i.d	HEX	R/O	HexRange R/O (0 .. 0)
i.e	HEX	R/O	HexRange R/O (0 .. 0)
i.f	HEX	R/O	HexRange R/O (0 .. 0)
i.10	HEX	R/O	HexRange R/O (0 .. 0)
i.11	HEX	R/O	HexRange R/O (0 .. 0)
i.12	HEX	R/O	HexRange R/O (0 .. 0)
i.13	HEX	R/O	HexRange R/O (0 .. 0)
i.14	HEX	R/O	HexRange R/O (0 .. 0)
i.15	HEX	R/O	HexRange R/O (0 .. 0)
i.16	HEX	R/O	HexRange R/O (0 .. 0)
i.17	HEX	R/O	HexRange R/O (0 .. 0)
i.18	HEX	R/O	HexRange R/O (0 .. 0)
i.19	HEX	R/O	HexRange R/O (0 .. 0)
i.1a	HEX	R/O	HexRange R/O (0 .. 0)
i.1b	HEX	R/O	HexRange R/O (0 .. 0)
i.1c	HEX	R/O	HexRange R/O (0 .. 0)
i.1d	HEX	R/O	HexRange R/O (0 .. 0)
i.1e	HEX	R/O	HexRange R/O (0 .. 0)
i.1f	HEX	R/O	HexRange R/O (0 .. 0)
i.q	HEX	R/O	HexRange R/O (0 .. 0)
l	TEXT	R/O	StringRange R/O

9.380 Moxa-ioLogik-4200

This page describes the device driver and the device window for the * Moxa ioLogik 4200* data

acquisition unit. The device driver monitors and controls the digital inputs/outputs of various modules hosted by the ioLogik 4200 unit. The driver supports digital I/O modules, the M-3802 current sensor module and the M6200 temperature measurement module (see below for details about the analog input modules).

The *Moxa-ioLogik-4200* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the *Moxa-ioLogik-4200* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Moxa-ioLogik-4000* device driver as a low level driver to access the physical I/O ports.

Port adressing

The ioLogik device must be configured to work in fixed address mapping mode to work with the driver. In this mode, each module has a fixed Modbus address range which can be evaluated from the module's position (its distance to the main module. The driver tests the addressing mode at startup, if the Moxa is configured to "DYNAMIC" addressing, a "Addressing mode" fault is raised. The driver supports up to 16 digital I/O modules at one device. The Moxa ioLogik 4200 main module limits the system to this size.

The sat-nms device driver for the Moxa ioLogik 4200 provides one input variable (named I0000 to I000F) and one output variable (named Q0000 to Q000F) for each module position. If for example an input module is at the third position from the main unit, its readings are in the variable I0002. The enumeration scheme (hexadecimal notation and starting with 0000 for the first one) is for compatibility with other PLC devices.

Output modules are treated in the same way but with the variables Q0000 Q000F. If for example the module next to the main module is an output or relay module, it gets controlled through the variable Q0000.

Analog readings

The sat-nms device driver for the Moxa ioLogik 4200 supports the usage of the M-6200 temperature measurement module and M-3802 8-channel current measurement module. It however limits the number of analog readings to 16 in total. This means, all analog modules installed may not provide more than 16 measurement values. If the driver detects more than 16 analog channels, it raises an "Analog channels" fault and ignores all analog reading beyond the limit.

The sat-nms device driver treats analog readings in a special way: There are 16 variables (named A0000 to A000F) provided which get the analog readings dynamically assigned. A0000 contains the first channel of the first module, A0001 the second channel. With the A???? variables not the absolute position of the module selects the variable but its relative position in the sequence of analog input modules.

If for example a M-3800 current sensor module is located next to the main module and the only

other analog input module in the system is a M-6200 located somewhere more to the right, the variables A0000 to A0007 contain the eight current readings of the M-3802 and the variables A0008 and A0009 the temperature readings of the M-6200.

Converting analog readings to readable format

The A???? variables contain the analog readings (temperature or current) in the native format as read from the device. This is a raw hexadecimal value of the ADC reading with a special coding for faults like broken cables.

For a temperature display in °C the sat-nms software provides the Moxa-ioLogik-Temperature logical device, which translates this value into a readable floating point number. The device also recognizes the special reading '0x8000' which is shown by the module to signal a broken cable to the temperature sensor and generates a fault flag from this. Analogously to this, the Moxa-ioLogik-Current logical device translates the ADC readings to a value in milliamperes.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Moxa ioLogik device acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the ioLogik device in the device setup window, there must not be defined any other devices beside this module for this interface.
- The ioLogik unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msec for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***

parameter	*description*
address	Must always be set to 1 if the device is controlled Modbus over a serial line. With Modbus/TCP the device address doesn't matter.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.addressing	TEXT	R/O	StringRange R/O
info.modules	TEXT	R/O	StringRange R/O
info.module.1	TEXT	R/O	StringRange R/O
info.module.2	TEXT	R/O	StringRange R/O
info.module.3	TEXT	R/O	StringRange R/O
info.module.4	TEXT	R/O	StringRange R/O
info.module.5	TEXT	R/O	StringRange R/O
info.module.6	TEXT	R/O	StringRange R/O
info.module.7	TEXT	R/O	StringRange R/O
info.module.8	TEXT	R/O	StringRange R/O
info.module.9	TEXT	R/O	StringRange R/O
info.module.10	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.module.11	TEXT	R/O	StringRange R/O
info.module.12	TEXT	R/O	StringRange R/O
info.module.13	TEXT	R/O	StringRange R/O
info.module.14	TEXT	R/O	StringRange R/O
info.module.15	TEXT	R/O	StringRange R/O
info.module.16	TEXT	R/O	StringRange R/O
I0000	HEX	R/O	HexRange R/O (0 .. 0)
I0001	HEX	R/O	HexRange R/O (0 .. 0)
I0002	HEX	R/O	HexRange R/O (0 .. 0)
I0003	HEX	R/O	HexRange R/O (0 .. 0)
I0004	HEX	R/O	HexRange R/O (0 .. 0)
I0005	HEX	R/O	HexRange R/O (0 .. 0)
I0006	HEX	R/O	HexRange R/O (0 .. 0)
I0007	HEX	R/O	HexRange R/O (0 .. 0)
I0008	HEX	R/O	HexRange R/O (0 .. 0)
I0009	HEX	R/O	HexRange R/O (0 .. 0)
I000A	HEX	R/O	HexRange R/O (0 .. 0)
I000B	HEX	R/O	HexRange R/O (0 .. 0)
I000C	HEX	R/O	HexRange R/O (0 .. 0)
I000D	HEX	R/O	HexRange R/O (0 .. 0)
I000E	HEX	R/O	HexRange R/O (0 .. 0)
I000F	HEX	R/O	HexRange R/O (0 .. 0)
A0000	HEX	R/O	HexRange R/O (0 .. 0)
A0001	HEX	R/O	HexRange R/O (0 .. 0)
A0002	HEX	R/O	HexRange R/O (0 .. 0)
A0003	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
A0004	HEX	R/O	HexRange R/O (0 .. 0)
A0005	HEX	R/O	HexRange R/O (0 .. 0)
A0006	HEX	R/O	HexRange R/O (0 .. 0)
A0007	HEX	R/O	HexRange R/O (0 .. 0)
A0008	HEX	R/O	HexRange R/O (0 .. 0)
A0009	HEX	R/O	HexRange R/O (0 .. 0)
A000A	HEX	R/O	HexRange R/O (0 .. 0)
A000B	HEX	R/O	HexRange R/O (0 .. 0)
A000C	HEX	R/O	HexRange R/O (0 .. 0)
A000D	HEX	R/O	HexRange R/O (0 .. 0)
A000E	HEX	R/O	HexRange R/O (0 .. 0)
A000F	HEX	R/O	HexRange R/O (0 .. 0)
Q0000	HEX		HexRange (0 .. 0)
Q0001	HEX		HexRange (0 .. 0)
Q0002	HEX		HexRange (0 .. 0)
Q0003	HEX		HexRange (0 .. 0)
Q0004	HEX		HexRange (0 .. 0)
Q0005	HEX		HexRange (0 .. 0)
Q0006	HEX		HexRange (0 .. 0)
Q0007	HEX		HexRange (0 .. 0)
Q0008	HEX		HexRange (0 .. 0)
Q0009	HEX		HexRange (0 .. 0)
Q000A	HEX		HexRange (0 .. 0)
Q000B	HEX		HexRange (0 .. 0)
Q000C	HEX		HexRange (0 .. 0)
Q000D	HEX		HexRange (0 .. 0)
Q000E	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
Q000F	HEX		HexRange (0 .. 0)
Y0000	HEX		HexRange (0 .. 0)
Y0001	HEX		HexRange (0 .. 0)
Y0002	HEX		HexRange (0 .. 0)
Y0003	HEX		HexRange (0 .. 0)
Y0004	HEX		HexRange (0 .. 0)
Y0005	HEX		HexRange (0 .. 0)
Y0006	HEX		HexRange (0 .. 0)
Y0007	HEX		HexRange (0 .. 0)
Y0008	HEX		HexRange (0 .. 0)
Y0009	HEX		HexRange (0 .. 0)
Y000A	HEX		HexRange (0 .. 0)
Y000B	HEX		HexRange (0 .. 0)
Y000C	HEX		HexRange (0 .. 0)
Y000D	HEX		HexRange (0 .. 0)
Y000E	HEX		HexRange (0 .. 0)
Y000F	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
i.a	HEX	R/O	HexRange R/O (0 .. 0)
i.b	HEX	R/O	HexRange R/O (0 .. 0)
i.c	HEX	R/O	HexRange R/O (0 .. 0)
i.d	HEX	R/O	HexRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Addressing mode)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Analog channels)

9.381 Moxa-ioLogik-Current

The *Moxa-ioLogik-Current* logical device converts the binary current value read from a Moxa M-3802 I/O module in an ioLogik 4000 or 4200 controller to a readable current reading.

The M-3801 module marks invalid measurements (broken cable) with the special value '0x8000'. The *Moxa-ioLogik-Current* device detects this value and raises a fault in this case. The current value shown in case of this fault is '0.0 mA'.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual current value
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window the parameter ID of the Moxa ioLogik device must be set which holds the binary current value.

parameter	***description***
currentId	Enter the ID of the parameter which represents the binary current value to convert. With a Moxa-ioLogik-4000 device this is one of the I???? or J???? variables of the device (see below). With a Moxa-ioLogik-4200 controller the analog readings are provided by the A???? variables of the device.

Module addresses with Moxa ioLogik 4000

The Moxa ioLogik 4000 I/O controllers dynamically assign the Modbus addresses for the connected modules, starting at 0 with the first module mounted nearest at the base module. Modbus registers are 16 bit wide, but because many I/O modules require only 8 or 4 bit, the ioLogik 4000 manages Modbus addresses in steps of 8 bit (half 16-bit registers). Hence, it can happen, that the current register of a M-3801 module may be split over two modbus registers.

To deal with this, the Moxa-ioLogik-4000 driver provides beside the "I????" variables which each hold the contents of one Modbus register a second set of variables called "J0000" to "J001E". These contain 16 bit values with 8-bit address shift. The "J????" variables may be used to read current values which are split across two Modbus addresses.

If e.g. the ioLogik Web GUI states, that the current register of your interest is at "0012", simply use the "I0012" variable in "currentId". If the ioLogik Web GUI states, that the current register is at "0012+08", use the "J0012" variable instead.

Module addresses with Moxa ioLogik 4200

The Moxa ioLogik 4200 I/O controller uses a different, static Modbus addressing scheme, however the sat-nms driver for this controller dynamically assigns analog readings to the parameter A0000 to A000F. A0000 contains the first channel of the first M-3802 module (that one next to the main module). A0001 contains the second channel of this module, A0007 the 8th channel of this module and so on. The sat-nms Moxa-ioLogik-4200 device driver supports up to 16 analog readings from various analog input modules. As the M-3802 module has 8 channels, up to two of these modules are supported in one device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.currentId	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Current sensor)
current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

9.382 Moxa-ioLogik-Temperature

The *Moxa-ioLogik-Temperature* logical device converts the binary temperature value read from a Moxa M-6200 I/O module in an ioLogik 4000 or 4200 controller to a readable temperature value. The device assumes, that the M-6200 module is configured to measure the temperature in centigrade.

The M-6200 module marks invalid measurements (broken cable) with the special value '0x8000'. The *Moxa-ioLogik-Temperature* device detects this value and raises a fault in this case. The temperature value shown in case of this fault is '0.0 °C'.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual temperature value
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window the parameter ID of the Moxa ioLogik device must be set which holds the binary temperature value.

parameter	*description*
temperatureId	Enter the ID of the parameter which represents the binary temperature value to convert. With a Moxa-ioLogik-4000 device this is one of the I???? or J???? variables of the device (see below). With a Moxa-ioLogik-4200 controller the analog readings are provided by the A???? variables of the device.

Module addresses with Moxa ioLogik 4000

The Moxa ioLogik 4000 I/O controllers dynamically assign the Modbus addresses for the connected modules, starting at 0 with the first module mounted nearest at the base module. Modbus registers are 16 bit wide, but because many I/O modules require only 8 or 4 bit, the ioLogik 4000 manages Modbus addresses in steps of 8 bit (half 16-bit registers). Hence, it can happen, that the temperature register of a M-6200 module may be split over two modbus registers.

To deal with this, the Moxa-ioLogik-4000 driver provides beside the "I????" variables which each hold the contents of one Modbus register a second set of variables called "J0000" to "J001E". These contain 16 bit values with 8-bit address shift. The "J????" variables may be used to read temperature values which are split across two Modbus addresses.

If e.g. the ioLogik Web GUI states, that the temperature register of your interest is at "0012", simply use the "I0012" variable in "temperatureId". If the ioLogik Web GUI states, that the temperature register is at "0012+08", use the "J0012" variable instead.

Module addresses with Moxa ioLogik 4200

The Moxa ioLogik 4200 I/O controller uses a different, static Modbus addressing scheme, however the sat-nms driver for this controller dynamically assigns analog readings to the parameter A0000 to A000F. A0000 contains the first channel of the first M-6200 module (that one next to the main module). A0001 contains the second channel of this module and so on. The sat-nms Moxa-ioLogik-4200 device driver supports up to 16 analog readings from various analog input modules.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.temperatureId	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature sensor)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

9.383 Moxa-ioThinx-4510

This page describes the device driver and the device window for the*Moxa ioThinx 4510* data acquisition unit. The device driver monitors and controls the digital inputs/outputs of various modules hosted by the ioLThinx 4510 unit. The driver supports digital I/O modules.

The *Moxa-ioThinx-4510* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*Moxa-ioThinx-4510* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *Moxa-ioThinx-4510* device driver as a low level driver to access the physical I/O ports.

Port adressing

As from units documentation there is no auto/fixed address mapping mode available the ioThinx device register addresses has to be manually added via device setup. See configuration parameters section below. By default all modules are disabled on the device setup page (module = NONE).

The driver was tested with MR1600 (16 DIs) and MR2404 (4 relay) modules. It is assumed that module base information can be found in the following registers

- SW Version: **47772**
- Unit Serial: **47780**
- Unit Name: **47744**

The sat-nms device driver for the Moxa ioThinx 4510 provides one input variable (named I0000 to I000F) and one output variable (named Q0000 to Q000F) for each module position. If for example an input module is at the third position from the main unit, its readings are in the variable I0002. The enumeration scheme (hexadecimal notation and starting with 0000 for the first one) is for compatibility with other PLC devices.

Output modules are treated in the same way but with the variables Q0000 Q000F. If for example the module next to the main module is an output or relay module, it gets controlled through the variable Q0000.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the Moxa ioLThinx device acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the ioThinx device in the device setup window, there must not be defined any other devices beside this module for this interface.
- The ioThinx unit is controlled through Modbus/TCP. The device's IP address must be set as an interface name like with serial devices controlled through a network terminal adapter: "aaa.bbb.ccc.ddd:502" where 502 is the standard Modbus/TCP port number.
- The IDLE time for the interface must be set to 200 msec for a fast response of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1 if the device is controlled Modbus over a serial line. With Modbus/TCP the device address doesn't matter.
module.01	The 1st module type to be controlled by I0000 or Q0000.
module.02	The 2nd module type to be controlled by I0001 or Q0001.
module.03	The 3rd module type to be controlled by I0002 or Q0002.
module.04	The 4th module type to be controlled by I0003 or Q0003.
module.05	The 5th module type to be controlled by I0004 or Q0004.
module.06	The 6th module type to be controlled by I0005 or Q0005.
module.07	The 7th module type to be controlled by I0006 or Q0006.

parameter	*description*
module.08	The 8th module type to be controlled by I0007 or Q0007.
module.09	The 9th module type to be controlled by I0008 or Q0008.
module.10	The 10th module type to be controlled by I0009 or Q0009.
module.11	The 11th module type to be controlled by I000A or Q000A.
module.12	The 12th module type to be controlled by I000B or Q000B.
module.13	The 13th module type to be controlled by I000C or Q000C.
module.14	The 14th module type to be controlled by I000D or Q000D.
module.15	The 15th module type to be controlled by I000E or Q000E.
module.16	The 16th module type to be controlled by I000F or Q000F.
address.01	The 1st module register address as shown in the WebUI.
address.02	The 2nd module register address as shown in the WebUI.
address.03	The 3rd module register address as shown in the WebUI.
address.04	The 4th module register address as shown in the WebUI.
address.05	The 5th module register address as shown in the WebUI.
address.06	The 6th module register address as shown in the WebUI.
address.07	The 7th module register address as shown in the WebUI.
address.08	The 8th module register address as shown in the WebUI.
address.09	The 9th module register address as shown in the WebUI.
address.10	The 10th module register address as shown in the WebUI.
address.11	The 11th module register address as shown in the WebUI.
address.12	The 12th module register address as shown in the WebUI.
address.13	The 13th module register address as shown in the WebUI.
address.14	The 14th module register address as shown in the WebUI.
address.15	The 15th module register address as shown in the WebUI.
address.16	The 16th module register address as shown in the WebUI.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.module.01	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.02	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.03	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.04	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.05	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.06	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.07	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.08	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.09	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.10	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.11	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.12	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.13	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.14	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.module.15	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)

name	*type*	*flags*	*range*
config.module.16	CHOICE	SAVE SETUP	EnumRange (NONE MR1600 MR2404)
config.address.01	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.02	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.03	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.04	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.05	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.06	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.07	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.08	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.09	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.address.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
I0000	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0001	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0002	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0003	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0004	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0005	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0006	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0007	HEX	R/O	HexRange R/O DIS (0 .. 0)
I0008	HEX	R/O	HexRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
I0009	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000A	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000B	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000C	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000D	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000E	HEX	R/O	HexRange R/O DIS (0 .. 0)
I000F	HEX	R/O	HexRange R/O DIS (0 .. 0)
Q0000	HEX		HexRange DIS (0 .. 0)
Q0001	HEX		HexRange DIS (0 .. 0)
Q0002	HEX		HexRange DIS (0 .. 0)
Q0003	HEX		HexRange DIS (0 .. 0)
Q0004	HEX		HexRange DIS (0 .. 0)
Q0005	HEX		HexRange DIS (0 .. 0)
Q0006	HEX		HexRange DIS (0 .. 0)
Q0007	HEX		HexRange DIS (0 .. 0)
Q0008	HEX		HexRange DIS (0 .. 0)
Q0009	HEX		HexRange DIS (0 .. 0)
Q000A	HEX		HexRange DIS (0 .. 0)
Q000B	HEX		HexRange DIS (0 .. 0)
Q000C	HEX		HexRange DIS (0 .. 0)
Q000D	HEX		HexRange DIS (0 .. 0)
Q000E	HEX		HexRange DIS (0 .. 0)
Q000F	HEX		HexRange DIS (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
i	HEX	R/O	HexRange R/O (0 .. 0)

9.384 Multi-Add

The logical Multi-Add device is intended to be used to add up to 32 parameters from various devices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual input values of the device and the overall sum as float and integer.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Arithmetic device with the devices that deliver the input signals and to configure the device's performance.

parameter	***description***
input.1-32-id	Enter the ID of the parameter the device shall use as input here. Leave the field empty if not used.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input.1.id	TEXT	SAVE SETUP	StringRange
config.input.2.id	TEXT	SAVE SETUP	StringRange
config.input.3.id	TEXT	SAVE SETUP	StringRange
config.input.4.id	TEXT	SAVE SETUP	StringRange
config.input.5.id	TEXT	SAVE SETUP	StringRange
config.input.6.id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input.7.id	TEXT	SAVE SETUP	StringRange
config.input.8.id	TEXT	SAVE SETUP	StringRange
config.input.9.id	TEXT	SAVE SETUP	StringRange
config.input.10.id	TEXT	SAVE SETUP	StringRange
config.input.11.id	TEXT	SAVE SETUP	StringRange
config.input.12.id	TEXT	SAVE SETUP	StringRange
config.input.13.id	TEXT	SAVE SETUP	StringRange
config.input.14.id	TEXT	SAVE SETUP	StringRange
config.input.15.id	TEXT	SAVE SETUP	StringRange
config.input.16.id	TEXT	SAVE SETUP	StringRange
config.input.17.id	TEXT	SAVE SETUP	StringRange
config.input.18.id	TEXT	SAVE SETUP	StringRange
config.input.19.id	TEXT	SAVE SETUP	StringRange
config.input.20.id	TEXT	SAVE SETUP	StringRange
config.input.21.id	TEXT	SAVE SETUP	StringRange
config.input.22.id	TEXT	SAVE SETUP	StringRange
config.input.23.id	TEXT	SAVE SETUP	StringRange
config.input.24.id	TEXT	SAVE SETUP	StringRange
config.input.25.id	TEXT	SAVE SETUP	StringRange
config.input.26.id	TEXT	SAVE SETUP	StringRange
config.input.27.id	TEXT	SAVE SETUP	StringRange
config.input.28.id	TEXT	SAVE SETUP	StringRange
config.input.29.id	TEXT	SAVE SETUP	StringRange
config.input.30.id	TEXT	SAVE SETUP	StringRange
config.input.31.id	TEXT	SAVE SETUP	StringRange
config.input.32.id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
internal.input.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.4	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.5	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.6	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.7	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.8	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.9	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.10	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.11	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.12	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.13	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.14	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.15	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.16	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.input.17	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.18	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.19	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.20	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.21	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.22	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.23	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.24	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.25	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.26	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.27	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.28	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.29	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.30	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.31	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.input.32	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
output	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
output.int	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.385 Multi-Protection-Switch

The Multi-Protection-Switch logical device implements up to 16 N:1 redundancy control units for up to 32 devices / chains in one logical device. The assignment of chains to a certain redundancy control may be done by the operator 'on the fly' in the running system.

While originally developed as a modem / modulator redundancy controller, the Multi-Protection-Switch device may be used for other purposes as well. It only requires that:

- Each device or device chain monitored by the device must provide one fault flag signalling any malfunction of the device. The sat-nms logical devices Logical-Gate or Logical-Gate-3 may be used to combine multiple fault conditions to the summary fault of the device chain.
- There must be one ON/OFF switch which switches the device chain in place (ON) or to standby (OFF). With satellite modulators this is the TX-ON switch, that's the reason why this parameter is labeled TX-ON in the device.
- There may be a signal input switch, which specially for the redundant chains control the input signal processed by this chain.

Configuration

The relevant parameter IDs of each device / device chain controlled by the device must be configured at the device's setup page. Once a device chain has a 'txOnId' set, it appears as configured at the 'Group definition' page and may be assigned to a 'group'.

A 'group' in this context means a N:1 protection switch unit. A group may consist of 2 to 32 device chains, exactly one of them must be configured as a spare chain.

The 'Group definition' page lets you change the group assignment of each chain on the fly in the running system. You are strongly encouraged to set the global protection enable to OFF while you modify the chain configuration. Unwanted automated switching may be triggered due to new / partly configured chains coming into a group.

Operation

Each group acts as a independent N:1 protection switch. If enabled, it replaces a faulty device by the spare. The switchover process is as follows:

1. The faulty device is switched TX-OFF.
2. The input switch of the spare device is set to the same input as the switch of the faulty device was set (if input switch IDs are configured).
3. All settings of the faulty device are copied to the spare device. As the settings memorized

in the device driver are used for this, copying works even if the faulty device if completely down.

4. The spare device is switched TX-ON. The redundancy switching is 'one way', the first fault that is recognized triggers a switchover, after this the protection switch has to be reset to enable it again. Resetting the protection before the fault condition has been cleared causes the switch to switchover again immediately.

Redundancy switch enabling

There are three levels of enabling the redundancy switch function of the device. First the primary page of the device window provides a global enable switch (at the very bottom of the page). Setting this to OFF inhibits any redundancy switching. The same page shows one enable switch for each group. Setting the enable parameter of a group to OFF inhibits the redundancy switching for this particular group. Then, at the 'Group definition' page, there is a enable switch for each individual device chain. By setting the enable switch of a device chain to OFF, you may exclude this chain from redundancy switching (This in fact masks the fault of this chain).

A fault of the spare device also inhibits redundancy switch for this protection switch. The Multi-Protection-Switch signals this as a fault.

Delay times

The protection switches are triggered by every fault condition, there is no delay / filtering forseen. If such a filtering is required to prevent short fault conditions from triggering a switchover, you may use the fault delay parameters in each individual device protected by the redundancy controller.

The Multi-Protection-Switch provides a 'tx on delay' which postpones the TX-ON command at the spare chain for some time (0..5000 msecs). This may be useful for devices which are busy for some time after parameter mirroring and therefore would ignore an immediate TX-ON command.

Input switch neutral position

For each group a input switch neutral position may defined at the setup page. If this switch position is not empty, the protection switch will change the input switch of a chain which caused a redundancy switch to neutral. The original switch position is remembered and will be restored when a 'RESET' is commanded for this protection switch. The redundant chain will be switched neutral with the 'RESET' command.

Using a neutral input switch position is helpful with setups using bi-directional modems. With a redundancy switchover not only the faulty chain is switched TX OFF, also the receive signal gets removed.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the actual state of each group and permits to reset or disable individual groups.
- --- This page shows the state of each device chain and lets you configure the group assignment for each chain.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
faultId.*	Defines the parameter ID of the fault flag which is monitored for each device chain. This may be a individual fault flag of a device, a summary fault flag of the device or a hardware fault input provided by an Input logical device.
txOnId.*	Defines the ID of the parameter which switches a device chain ON/OFF. The device name contained in this ID defines the device which will be 'mirrored' during switchover.
switchId.*	Defines the parameter ID for a switch selecting the input signal of a device chain. Setting this ID is optional, but if set, it must be set for each chain within a group.
neutral.*	Defines the neutral input switch position for each group. The handling of neutral switch positions is described above at paragraph 'Input switch neutral position'

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.faultId.1	TEXT	SAVE SETUP	StringRange
config.faultId.2	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.faultId.3	TEXT	SAVE SETUP	StringRange
config.faultId.4	TEXT	SAVE SETUP	StringRange
config.faultId.5	TEXT	SAVE SETUP	StringRange
config.faultId.6	TEXT	SAVE SETUP	StringRange
config.faultId.7	TEXT	SAVE SETUP	StringRange
config.faultId.8	TEXT	SAVE SETUP	StringRange
config.faultId.9	TEXT	SAVE SETUP	StringRange
config.faultId.10	TEXT	SAVE SETUP	StringRange
config.faultId.11	TEXT	SAVE SETUP	StringRange
config.faultId.12	TEXT	SAVE SETUP	StringRange
config.faultId.13	TEXT	SAVE SETUP	StringRange
config.faultId.14	TEXT	SAVE SETUP	StringRange
config.faultId.15	TEXT	SAVE SETUP	StringRange
config.faultId.16	TEXT	SAVE SETUP	StringRange
config.faultId.17	TEXT	SAVE SETUP	StringRange
config.faultId.18	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.faultId.19	TEXT	SAVE SETUP	StringRange
config.faultId.20	TEXT	SAVE SETUP	StringRange
config.faultId.21	TEXT	SAVE SETUP	StringRange
config.faultId.22	TEXT	SAVE SETUP	StringRange
config.faultId.23	TEXT	SAVE SETUP	StringRange
config.faultId.24	TEXT	SAVE SETUP	StringRange
config.faultId.25	TEXT	SAVE SETUP	StringRange
config.faultId.26	TEXT	SAVE SETUP	StringRange
config.faultId.27	TEXT	SAVE SETUP	StringRange
config.faultId.28	TEXT	SAVE SETUP	StringRange
config.faultId.29	TEXT	SAVE SETUP	StringRange
config.faultId.30	TEXT	SAVE SETUP	StringRange
config.faultId.31	TEXT	SAVE SETUP	StringRange
config.faultId.32	TEXT	SAVE SETUP	StringRange
config.txOnId.1	TEXT	SAVE SETUP	StringRange
config.txOnId.2	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.txOnId.3	TEXT	SAVE SETUP	StringRange
config.txOnId.4	TEXT	SAVE SETUP	StringRange
config.txOnId.5	TEXT	SAVE SETUP	StringRange
config.txOnId.6	TEXT	SAVE SETUP	StringRange
config.txOnId.7	TEXT	SAVE SETUP	StringRange
config.txOnId.8	TEXT	SAVE SETUP	StringRange
config.txOnId.9	TEXT	SAVE SETUP	StringRange
config.txOnId.10	TEXT	SAVE SETUP	StringRange
config.txOnId.11	TEXT	SAVE SETUP	StringRange
config.txOnId.12	TEXT	SAVE SETUP	StringRange
config.txOnId.13	TEXT	SAVE SETUP	StringRange
config.txOnId.14	TEXT	SAVE SETUP	StringRange
config.txOnId.15	TEXT	SAVE SETUP	StringRange
config.txOnId.16	TEXT	SAVE SETUP	StringRange
config.txOnId.17	TEXT	SAVE SETUP	StringRange
config.txOnId.18	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.txOnId.19	TEXT	SAVE SETUP	StringRange
config.txOnId.20	TEXT	SAVE SETUP	StringRange
config.txOnId.21	TEXT	SAVE SETUP	StringRange
config.txOnId.22	TEXT	SAVE SETUP	StringRange
config.txOnId.23	TEXT	SAVE SETUP	StringRange
config.txOnId.24	TEXT	SAVE SETUP	StringRange
config.txOnId.25	TEXT	SAVE SETUP	StringRange
config.txOnId.26	TEXT	SAVE SETUP	StringRange
config.txOnId.27	TEXT	SAVE SETUP	StringRange
config.txOnId.28	TEXT	SAVE SETUP	StringRange
config.txOnId.29	TEXT	SAVE SETUP	StringRange
config.txOnId.30	TEXT	SAVE SETUP	StringRange
config.txOnId.31	TEXT	SAVE SETUP	StringRange
config.txOnId.32	TEXT	SAVE SETUP	StringRange
config.switchId.1	TEXT	SAVE SETUP	StringRange
config.switchId.2	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.switchId.3	TEXT	SAVE SETUP	StringRange
config.switchId.4	TEXT	SAVE SETUP	StringRange
config.switchId.5	TEXT	SAVE SETUP	StringRange
config.switchId.6	TEXT	SAVE SETUP	StringRange
config.switchId.7	TEXT	SAVE SETUP	StringRange
config.switchId.8	TEXT	SAVE SETUP	StringRange
config.switchId.9	TEXT	SAVE SETUP	StringRange
config.switchId.10	TEXT	SAVE SETUP	StringRange
config.switchId.11	TEXT	SAVE SETUP	StringRange
config.switchId.12	TEXT	SAVE SETUP	StringRange
config.switchId.13	TEXT	SAVE SETUP	StringRange
config.switchId.14	TEXT	SAVE SETUP	StringRange
config.switchId.15	TEXT	SAVE SETUP	StringRange
config.switchId.16	TEXT	SAVE SETUP	StringRange
config.switchId.17	TEXT	SAVE SETUP	StringRange
config.switchId.18	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.switchId.19	TEXT	SAVE SETUP	StringRange
config.switchId.20	TEXT	SAVE SETUP	StringRange
config.switchId.21	TEXT	SAVE SETUP	StringRange
config.switchId.22	TEXT	SAVE SETUP	StringRange
config.switchId.23	TEXT	SAVE SETUP	StringRange
config.switchId.24	TEXT	SAVE SETUP	StringRange
config.switchId.25	TEXT	SAVE SETUP	StringRange
config.switchId.26	TEXT	SAVE SETUP	StringRange
config.switchId.27	TEXT	SAVE SETUP	StringRange
config.switchId.28	TEXT	SAVE SETUP	StringRange
config.switchId.29	TEXT	SAVE SETUP	StringRange
config.switchId.30	TEXT	SAVE SETUP	StringRange
config.switchId.31	TEXT	SAVE SETUP	StringRange
config.switchId.32	TEXT	SAVE SETUP	StringRange
config.neutral.1	TEXT	SAVE SETUP	StringRange
config.neutral.2	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.neutral.3	TEXT	SAVE SETUP	StringRange
config.neutral.4	TEXT	SAVE SETUP	StringRange
config.neutral.5	TEXT	SAVE SETUP	StringRange
config.neutral.6	TEXT	SAVE SETUP	StringRange
config.neutral.7	TEXT	SAVE SETUP	StringRange
config.neutral.8	TEXT	SAVE SETUP	StringRange
config.neutral.9	TEXT	SAVE SETUP	StringRange
config.neutral.10	TEXT	SAVE SETUP	StringRange
config.neutral.11	TEXT	SAVE SETUP	StringRange
config.neutral.12	TEXT	SAVE SETUP	StringRange
config.neutral.13	TEXT	SAVE SETUP	StringRange
config.neutral.14	TEXT	SAVE SETUP	StringRange
config.neutral.15	TEXT	SAVE SETUP	StringRange
config.neutral.16	TEXT	SAVE SETUP	StringRange
chain.group.1	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.2	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
chain.group.3	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.4	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.5	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.6	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.7	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.8	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.9	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.10	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.11	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.12	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.13	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.14	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.15	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.16	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.17	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.18	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
chain.group.19	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.20	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.21	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.22	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.23	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.24	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.25	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.26	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.27	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.28	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.29	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.30	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.31	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.group.32	CHOICE	SAVE	EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
chain.enable.1	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.2	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.3	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.4	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
chain.enable.5	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.6	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.7	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.8	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.9	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.10	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.11	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.12	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.13	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.14	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.15	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.16	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.17	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.18	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.19	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.20	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.21	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.22	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.23	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.24	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.25	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.26	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.27	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.28	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.29	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.30	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
chain.enable.31	CHOICE	SAVE	EnumRange (OFF ON)
chain.enable.32	CHOICE	SAVE	EnumRange (OFF ON)
chain.spare.1	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.2	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.3	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.4	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.5	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.6	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.7	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.8	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.9	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.10	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.11	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.12	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.13	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.14	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.15	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.16	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.17	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.18	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.19	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.20	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.21	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.22	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.23	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.24	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.25	CHOICE	SAVE	EnumRange (NRM SPARE)

name	*type*	*flags*	*range*
chain.spare.26	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.27	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.28	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.29	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.30	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.31	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.spare.32	CHOICE	SAVE	EnumRange (NRM SPARE)
chain.savedpos.1	TEXT	SAVE	StringRange
chain.savedpos.2	TEXT	SAVE	StringRange
chain.savedpos.3	TEXT	SAVE	StringRange
chain.savedpos.4	TEXT	SAVE	StringRange
chain.savedpos.5	TEXT	SAVE	StringRange
chain.savedpos.6	TEXT	SAVE	StringRange
chain.savedpos.7	TEXT	SAVE	StringRange
chain.savedpos.8	TEXT	SAVE	StringRange
chain.savedpos.9	TEXT	SAVE	StringRange
chain.savedpos.10	TEXT	SAVE	StringRange
chain.savedpos.11	TEXT	SAVE	StringRange
chain.savedpos.12	TEXT	SAVE	StringRange
chain.savedpos.13	TEXT	SAVE	StringRange
chain.savedpos.14	TEXT	SAVE	StringRange
chain.savedpos.15	TEXT	SAVE	StringRange
chain.savedpos.16	TEXT	SAVE	StringRange
chain.savedpos.17	TEXT	SAVE	StringRange
chain.savedpos.18	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
chain.savedpos.19	TEXT	SAVE	StringRange
chain.savedpos.20	TEXT	SAVE	StringRange
chain.savedpos.21	TEXT	SAVE	StringRange
chain.savedpos.22	TEXT	SAVE	StringRange
chain.savedpos.23	TEXT	SAVE	StringRange
chain.savedpos.24	TEXT	SAVE	StringRange
chain.savedpos.25	TEXT	SAVE	StringRange
chain.savedpos.26	TEXT	SAVE	StringRange
chain.savedpos.27	TEXT	SAVE	StringRange
chain.savedpos.28	TEXT	SAVE	StringRange
chain.savedpos.29	TEXT	SAVE	StringRange
chain.savedpos.30	TEXT	SAVE	StringRange
chain.savedpos.31	TEXT	SAVE	StringRange
chain.savedpos.32	TEXT	SAVE	StringRange
chain.configured.1	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.2	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.3	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.4	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.5	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.6	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.7	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.8	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.9	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.10	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.11	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.12	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
chain.configured.13	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.14	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.15	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.16	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.17	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.18	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.19	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.20	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.21	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.22	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.23	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.24	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.25	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.26	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.27	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.28	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.29	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.30	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.31	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.configured.32	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.name.1	TEXT	R/O	StringRange R/O
chain.name.2	TEXT	R/O	StringRange R/O
chain.name.3	TEXT	R/O	StringRange R/O
chain.name.4	TEXT	R/O	StringRange R/O
chain.name.5	TEXT	R/O	StringRange R/O
chain.name.6	TEXT	R/O	StringRange R/O
chain.name.7	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
chain.name.8	TEXT	R/O	StringRange R/O
chain.name.9	TEXT	R/O	StringRange R/O
chain.name.10	TEXT	R/O	StringRange R/O
chain.name.11	TEXT	R/O	StringRange R/O
chain.name.12	TEXT	R/O	StringRange R/O
chain.name.13	TEXT	R/O	StringRange R/O
chain.name.14	TEXT	R/O	StringRange R/O
chain.name.15	TEXT	R/O	StringRange R/O
chain.name.16	TEXT	R/O	StringRange R/O
chain.name.17	TEXT	R/O	StringRange R/O
chain.name.18	TEXT	R/O	StringRange R/O
chain.name.19	TEXT	R/O	StringRange R/O
chain.name.20	TEXT	R/O	StringRange R/O
chain.name.21	TEXT	R/O	StringRange R/O
chain.name.22	TEXT	R/O	StringRange R/O
chain.name.23	TEXT	R/O	StringRange R/O
chain.name.24	TEXT	R/O	StringRange R/O
chain.name.25	TEXT	R/O	StringRange R/O
chain.name.26	TEXT	R/O	StringRange R/O
chain.name.27	TEXT	R/O	StringRange R/O
chain.name.28	TEXT	R/O	StringRange R/O
chain.name.29	TEXT	R/O	StringRange R/O
chain.name.30	TEXT	R/O	StringRange R/O
chain.name.31	TEXT	R/O	StringRange R/O
chain.name.32	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
chain.fault.1	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.2	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.3	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.4	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.5	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.6	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.7	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.8	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.9	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.10	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.11	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.12	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.13	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.14	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.15	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.16	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.17	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.18	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.19	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.20	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.21	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.22	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.23	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.24	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.25	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.26	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.27	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
chain.fault.28	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.29	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.30	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.31	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.fault.32	BOOLEAN	R/O	BooleanRange R/O (true false)
group.name.1	TEXT	R/O	StringRange R/O
group.name.2	TEXT	R/O	StringRange R/O
group.name.3	TEXT	R/O	StringRange R/O
group.name.4	TEXT	R/O	StringRange R/O
group.name.5	TEXT	R/O	StringRange R/O
group.name.6	TEXT	R/O	StringRange R/O
group.name.7	TEXT	R/O	StringRange R/O
group.name.8	TEXT	R/O	StringRange R/O
group.name.9	TEXT	R/O	StringRange R/O
group.name.10	TEXT	R/O	StringRange R/O
group.name.11	TEXT	R/O	StringRange R/O
group.name.12	TEXT	R/O	StringRange R/O
group.name.13	TEXT	R/O	StringRange R/O
group.name.14	TEXT	R/O	StringRange R/O
group.name.15	TEXT	R/O	StringRange R/O
group.name.16	TEXT	R/O	StringRange R/O
group.enable.1	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.2	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.3	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.4	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.5	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
group.enable.6	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.7	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.8	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.9	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.10	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.11	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.12	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.13	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.14	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.15	CHOICE	SAVE	EnumRange (OFF ON)
group.enable.16	CHOICE	SAVE	EnumRange (OFF ON)
group.reset.1	TEXT	NOPRESET	StringRange
group.reset.2	TEXT	NOPRESET	StringRange
group.reset.3	TEXT	NOPRESET	StringRange
group.reset.4	TEXT	NOPRESET	StringRange
group.reset.5	TEXT	NOPRESET	StringRange
group.reset.6	TEXT	NOPRESET	StringRange
group.reset.7	TEXT	NOPRESET	StringRange
group.reset.8	TEXT	NOPRESET	StringRange
group.reset.9	TEXT	NOPRESET	StringRange
group.reset.10	TEXT	NOPRESET	StringRange
group.reset.11	TEXT	NOPRESET	StringRange
group.reset.12	TEXT	NOPRESET	StringRange
group.reset.13	TEXT	NOPRESET	StringRange
group.reset.14	TEXT	NOPRESET	StringRange
group.reset.15	TEXT	NOPRESET	StringRange

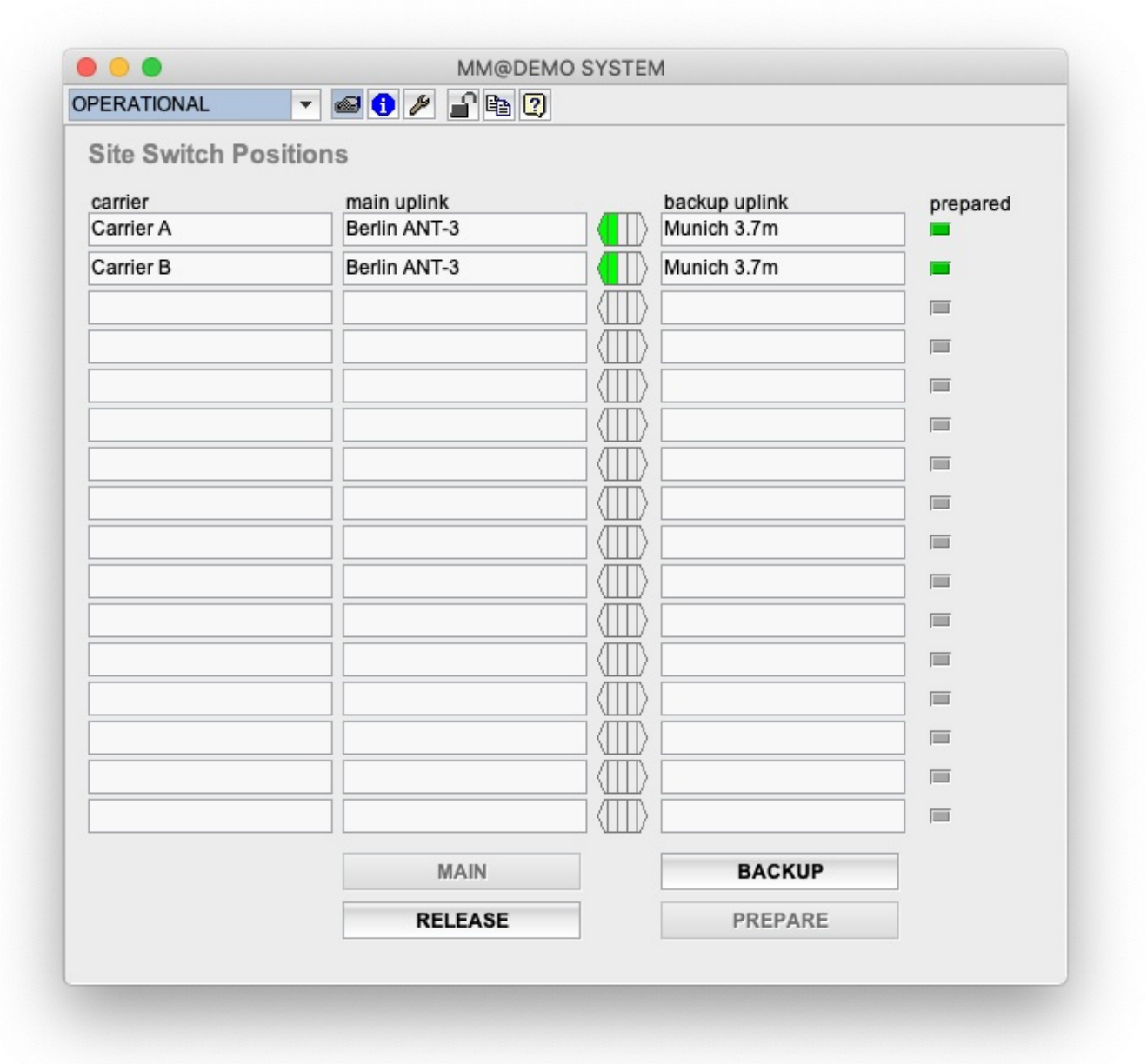
name	*type*	*flags*	*range*
group.reset.16	TEXT	NOPRESET	StringRange
group.state.1	TEXT	R/O	StringRange R/O
group.state.2	TEXT	R/O	StringRange R/O
group.state.3	TEXT	R/O	StringRange R/O
group.state.4	TEXT	R/O	StringRange R/O
group.state.5	TEXT	R/O	StringRange R/O
group.state.6	TEXT	R/O	StringRange R/O
group.state.7	TEXT	R/O	StringRange R/O
group.state.8	TEXT	R/O	StringRange R/O
group.state.9	TEXT	R/O	StringRange R/O
group.state.10	TEXT	R/O	StringRange R/O
group.state.11	TEXT	R/O	StringRange R/O
group.state.12	TEXT	R/O	StringRange R/O
group.state.13	TEXT	R/O	StringRange R/O
group.state.14	TEXT	R/O	StringRange R/O
group.state.15	TEXT	R/O	StringRange R/O
group.state.16	TEXT	R/O	StringRange R/O
group.nstat.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.9	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
group.nstat.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.nstat.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
group.configured.1	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.2	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.3	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.4	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.5	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.6	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.7	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.8	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.9	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.10	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.11	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.12	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.13	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.14	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.15	BOOLEAN	R/O	BooleanRange R/O (true false)
group.configured.16	BOOLEAN	R/O	BooleanRange R/O (true false)
txOnDelay	INTEGER	SAVE	IntegerRange (0 .. 5000)
globalEnable	CHOICE	SAVE	EnumRange (OFF ON)

9.386 Multi-Site-Switch

The Multi-Site-Switch logical device is used to switch multiple carriers from a primary uplink site to a backup site synchronously an vice versa. The Multi-Site-Switch device does this by monitoring and controlling a set of Carrier-Site-Switch devices, showing the actual state of these devices and doing the switchover synchronously.

The Multi-Site-Switch device may be configured to control up to 16 Carrier-Site-Switch devices, the devices may be added or removed on the fly without restarting the software.



Operating the device is quite simple: Once you have set the Carrier-Site-Switch device which shall be controlled at the maintenance page of the device, they appear with their actual states at the Multi-Site-Switch device's main page as show above. Below the list of carriers to switch,

there are the same four buttons as with the Carrier-Site-Switch device to prepare / release the backup site and to switch between the sites.

The Multi-Site-Switch device requires all controlled devices to be in the same initial state. If this is not the case, you cannot operate the Multi-Site-Switch. To resolve this situation, you should go to each single Carrier-Site-Switch device, setting them all to transmit either from the main or the backup site.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands actual uplink locations.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the devices that will perform the carrier on/off settings.

parameter	***description***
csw.1 .. csw.16	The device names of the Carrier-Site-Switch devices to control.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.csw.1	TEXT	SAVE SETUP	StringRange
config.csw.2	TEXT	SAVE SETUP	StringRange
config.csw.3	TEXT	SAVE SETUP	StringRange
config.csw.4	TEXT	SAVE SETUP	StringRange
config.csw.5	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.csw.6	TEXT	SAVE SETUP	StringRange
config.csw.7	TEXT	SAVE SETUP	StringRange
config.csw.8	TEXT	SAVE SETUP	StringRange
config.csw.9	TEXT	SAVE SETUP	StringRange
config.csw.10	TEXT	SAVE SETUP	StringRange
config.csw.11	TEXT	SAVE SETUP	StringRange
config.csw.12	TEXT	SAVE SETUP	StringRange
config.csw.13	TEXT	SAVE SETUP	StringRange
config.csw.14	TEXT	SAVE SETUP	StringRange
config.csw.15	TEXT	SAVE SETUP	StringRange
config.csw.16	TEXT	SAVE SETUP	StringRange
main	TEXT		StringRange
bckp	TEXT		StringRange
prepare	TEXT		StringRange
release	TEXT		StringRange
progressReport.1	TEXT	R/O	StringRange R/O
progressIcon.1	TEXT	R/O	StringRange R/O
prepared.1	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.1	BOOLEAN	R/O	BooleanRange R/O (true false)
location.1	TEXT	R/O	StringRange R/O
mainName.1	TEXT	R/O	StringRange R/O
mainDescription.1	TEXT	R/O	StringRange R/O
bckpName.1	TEXT	R/O	StringRange R/O
bckpDescription.1	TEXT	R/O	StringRange R/O
comment.1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
progressReport.2	TEXT	R/O	StringRange R/O
progressIcon.2	TEXT	R/O	StringRange R/O
prepared.2	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.2	BOOLEAN	R/O	BooleanRange R/O (true false)
location.2	TEXT	R/O	StringRange R/O
mainName.2	TEXT	R/O	StringRange R/O
mainDescription.2	TEXT	R/O	StringRange R/O
bckpName.2	TEXT	R/O	StringRange R/O
bckpDescription.2	TEXT	R/O	StringRange R/O
comment.2	TEXT	R/O	StringRange R/O
progressReport.3	TEXT	R/O	StringRange R/O
progressIcon.3	TEXT	R/O	StringRange R/O
prepared.3	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.3	BOOLEAN	R/O	BooleanRange R/O (true false)
location.3	TEXT	R/O	StringRange R/O
mainName.3	TEXT	R/O	StringRange R/O
mainDescription.3	TEXT	R/O	StringRange R/O
bckpName.3	TEXT	R/O	StringRange R/O
bckpDescription.3	TEXT	R/O	StringRange R/O
comment.3	TEXT	R/O	StringRange R/O
progressReport.4	TEXT	R/O	StringRange R/O
progressIcon.4	TEXT	R/O	StringRange R/O
prepared.4	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
enabled.4	BOOLEAN	R/O	BooleanRange R/O (true false)
location.4	TEXT	R/O	StringRange R/O
mainName.4	TEXT	R/O	StringRange R/O
mainDescription.4	TEXT	R/O	StringRange R/O
bckpName.4	TEXT	R/O	StringRange R/O
bckpDescription.4	TEXT	R/O	StringRange R/O
comment.4	TEXT	R/O	StringRange R/O
progressReport.5	TEXT	R/O	StringRange R/O
progressIcon.5	TEXT	R/O	StringRange R/O
prepared.5	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.5	BOOLEAN	R/O	BooleanRange R/O (true false)
location.5	TEXT	R/O	StringRange R/O
mainName.5	TEXT	R/O	StringRange R/O
mainDescription.5	TEXT	R/O	StringRange R/O
bckpName.5	TEXT	R/O	StringRange R/O
bckpDescription.5	TEXT	R/O	StringRange R/O
comment.5	TEXT	R/O	StringRange R/O
progressReport.6	TEXT	R/O	StringRange R/O
progressIcon.6	TEXT	R/O	StringRange R/O
prepared.6	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.6	BOOLEAN	R/O	BooleanRange R/O (true false)
location.6	TEXT	R/O	StringRange R/O
mainName.6	TEXT	R/O	StringRange R/O
mainDescription.6	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
bckpName.6	TEXT	R/O	StringRange R/O
bckpDescription.6	TEXT	R/O	StringRange R/O
comment.6	TEXT	R/O	StringRange R/O
progressReport.7	TEXT	R/O	StringRange R/O
progressIcon.7	TEXT	R/O	StringRange R/O
prepared.7	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.7	BOOLEAN	R/O	BooleanRange R/O (true false)
location.7	TEXT	R/O	StringRange R/O
mainName.7	TEXT	R/O	StringRange R/O
mainDescription.7	TEXT	R/O	StringRange R/O
bckpName.7	TEXT	R/O	StringRange R/O
bckpDescription.7	TEXT	R/O	StringRange R/O
comment.7	TEXT	R/O	StringRange R/O
progressReport.8	TEXT	R/O	StringRange R/O
progressIcon.8	TEXT	R/O	StringRange R/O
prepared.8	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.8	BOOLEAN	R/O	BooleanRange R/O (true false)
location.8	TEXT	R/O	StringRange R/O
mainName.8	TEXT	R/O	StringRange R/O
mainDescription.8	TEXT	R/O	StringRange R/O
bckpName.8	TEXT	R/O	StringRange R/O
bckpDescription.8	TEXT	R/O	StringRange R/O
comment.8	TEXT	R/O	StringRange R/O
progressReport.9	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
progressIcon.9	TEXT	R/O	StringRange R/O
prepared.9	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.9	BOOLEAN	R/O	BooleanRange R/O (true false)
location.9	TEXT	R/O	StringRange R/O
mainName.9	TEXT	R/O	StringRange R/O
mainDescription.9	TEXT	R/O	StringRange R/O
bckpName.9	TEXT	R/O	StringRange R/O
bckpDescription.9	TEXT	R/O	StringRange R/O
comment.9	TEXT	R/O	StringRange R/O
progressReport.10	TEXT	R/O	StringRange R/O
progressIcon.10	TEXT	R/O	StringRange R/O
prepared.10	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.10	BOOLEAN	R/O	BooleanRange R/O (true false)
location.10	TEXT	R/O	StringRange R/O
mainName.10	TEXT	R/O	StringRange R/O
mainDescription.10	TEXT	R/O	StringRange R/O
bckpName.10	TEXT	R/O	StringRange R/O
bckpDescription.10	TEXT	R/O	StringRange R/O
comment.10	TEXT	R/O	StringRange R/O
progressReport.11	TEXT	R/O	StringRange R/O
progressIcon.11	TEXT	R/O	StringRange R/O
prepared.11	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.11	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
location.11	TEXT	R/O	StringRange R/O
mainName.11	TEXT	R/O	StringRange R/O
mainDescription.11	TEXT	R/O	StringRange R/O
bckpName.11	TEXT	R/O	StringRange R/O
bckpDescription.11	TEXT	R/O	StringRange R/O
comment.11	TEXT	R/O	StringRange R/O
progressReport.12	TEXT	R/O	StringRange R/O
progressIcon.12	TEXT	R/O	StringRange R/O
prepared.12	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.12	BOOLEAN	R/O	BooleanRange R/O (true false)
location.12	TEXT	R/O	StringRange R/O
mainName.12	TEXT	R/O	StringRange R/O
mainDescription.12	TEXT	R/O	StringRange R/O
bckpName.12	TEXT	R/O	StringRange R/O
bckpDescription.12	TEXT	R/O	StringRange R/O
comment.12	TEXT	R/O	StringRange R/O
progressReport.13	TEXT	R/O	StringRange R/O
progressIcon.13	TEXT	R/O	StringRange R/O
prepared.13	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.13	BOOLEAN	R/O	BooleanRange R/O (true false)
location.13	TEXT	R/O	StringRange R/O
mainName.13	TEXT	R/O	StringRange R/O
mainDescription.13	TEXT	R/O	StringRange R/O
bckpName.13	TEXT	R/O	StringRange R/O

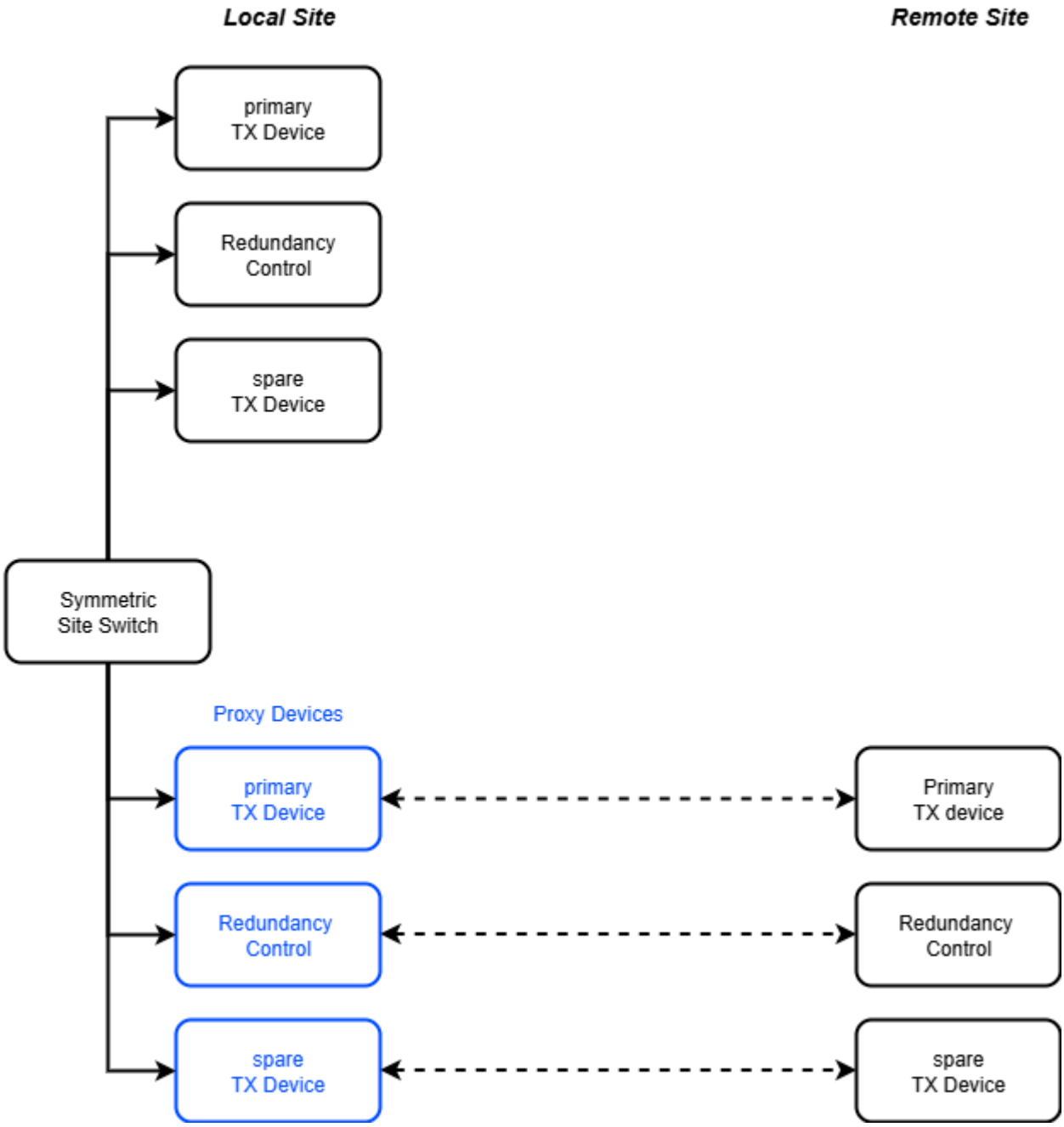
name	*type*	*flags*	*range*
bckpDescription.13	TEXT	R/O	StringRange R/O
comment.13	TEXT	R/O	StringRange R/O
progressReport.14	TEXT	R/O	StringRange R/O
progressIcon.14	TEXT	R/O	StringRange R/O
prepared.14	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.14	BOOLEAN	R/O	BooleanRange R/O (true false)
location.14	TEXT	R/O	StringRange R/O
mainName.14	TEXT	R/O	StringRange R/O
mainDescription.14	TEXT	R/O	StringRange R/O
bckpName.14	TEXT	R/O	StringRange R/O
bckpDescription.14	TEXT	R/O	StringRange R/O
comment.14	TEXT	R/O	StringRange R/O
progressReport.15	TEXT	R/O	StringRange R/O
progressIcon.15	TEXT	R/O	StringRange R/O
prepared.15	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.15	BOOLEAN	R/O	BooleanRange R/O (true false)
location.15	TEXT	R/O	StringRange R/O
mainName.15	TEXT	R/O	StringRange R/O
mainDescription.15	TEXT	R/O	StringRange R/O
bckpName.15	TEXT	R/O	StringRange R/O
bckpDescription.15	TEXT	R/O	StringRange R/O
comment.15	TEXT	R/O	StringRange R/O
progressReport.16	TEXT	R/O	StringRange R/O
progressIcon.16	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
prepared.16	BOOLEAN	R/O	BooleanRange R/O (true false)
enabled.16	BOOLEAN	R/O	BooleanRange R/O (true false)
location.16	TEXT	R/O	StringRange R/O
mainName.16	TEXT	R/O	StringRange R/O
mainDescription.16	TEXT	R/O	StringRange R/O
bckpName.16	TEXT	R/O	StringRange R/O
bckpDescription.16	TEXT	R/O	StringRange R/O
comment.16	TEXT	R/O	StringRange R/O

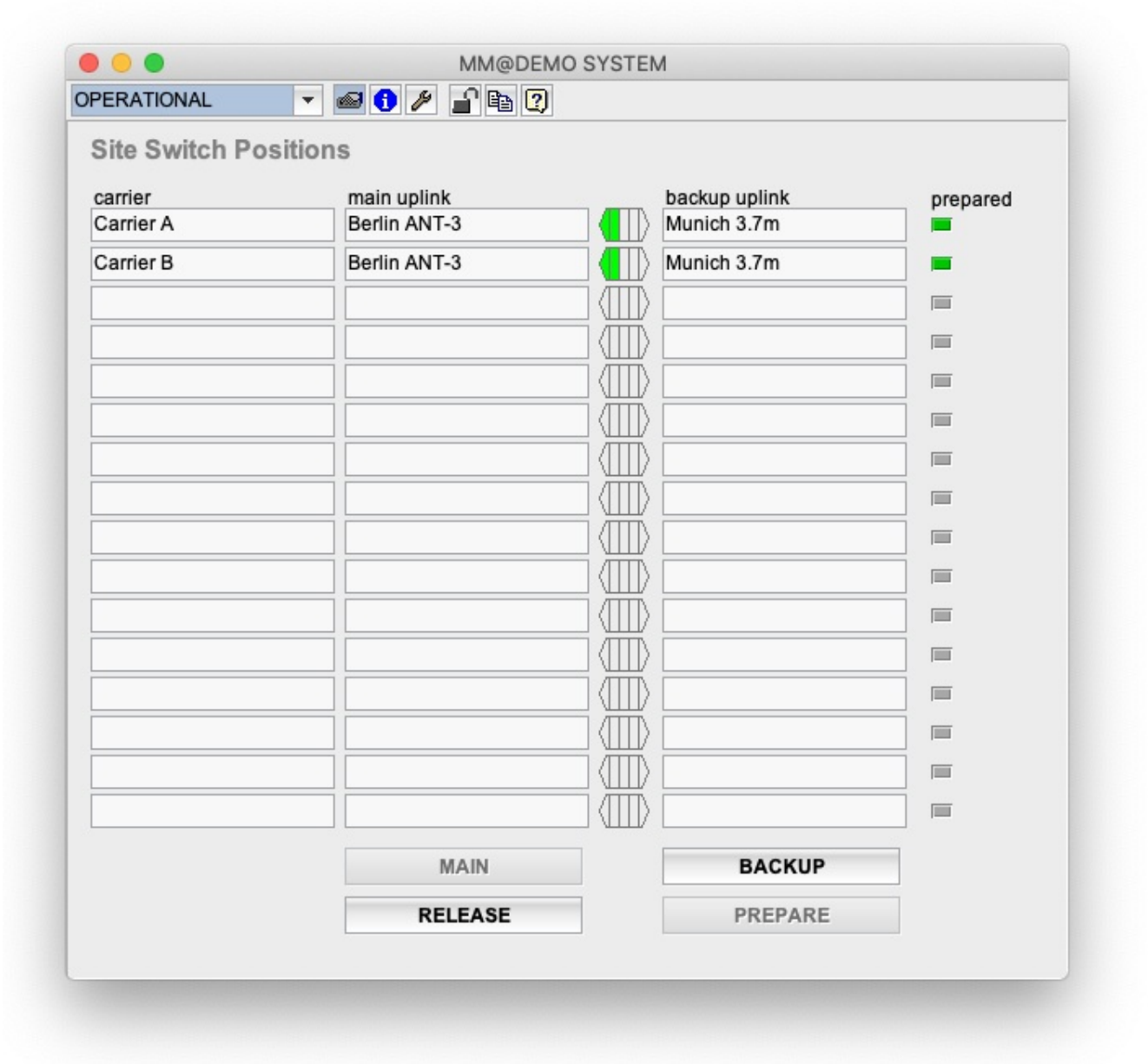
9.387 Multi-Symmetric-Site-Switch

The Multi-Site-Switch logical device is used to switch multiple carriers between two uplink locations. It is used together with the Symmetric-Site-Switch logical device.

Such devices control the equipment of one transmit chain for a site diversity application between two uplink ground stations. As the driver name implies, the device works completely symmetrically, there is no preference for a primary or a backup site.



The Multi-Site-Switch device may be configured to control up to 16 Symmetric-Site-Switch devices, the devices may be added or removed on the fly without restarting the software.



Operating the device is quite simple: Once you have set the Carrier-Site-Switch device which shall be controlled at the maintenance page of the device, they appear with their actual states at the Multi-Site-Switch device's main page as show above. Below the list of carriers to switch, there are the same buttons as with the Symmetric-Site-Switch device to switch between the sites.

The Multi-Site-Switch device requires all controlled devices to be in the same initial state. If this is not the case, you cannot operate the Multi-Site-Switch. To resolve this situation, you should go to each single Symmetric-Site-Switch device, setting them all to transmit either from the main or the backup site.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows.

- --- Shows / commands actual uplink locations.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the devices that will perform the carrier on/off settings.

parameter	***description***
csw.1 .. csw.16	The device names of the Symmetric-Site-Switch devices to control.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.ssw.1	TEXT	SAVE SETUP	StringRange
config.ssw.2	TEXT	SAVE SETUP	StringRange
config.ssw.3	TEXT	SAVE SETUP	StringRange
config.ssw.4	TEXT	SAVE SETUP	StringRange
config.ssw.5	TEXT	SAVE SETUP	StringRange
config.ssw.6	TEXT	SAVE SETUP	StringRange
config.ssw.7	TEXT	SAVE SETUP	StringRange
config.ssw.8	TEXT	SAVE SETUP	StringRange
config.ssw.9	TEXT	SAVE SETUP	StringRange
config.ssw.10	TEXT	SAVE SETUP	StringRange
config.ssw.11	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ssw.12	TEXT	SAVE SETUP	StringRange
config.ssw.13	TEXT	SAVE SETUP	StringRange
config.ssw.14	TEXT	SAVE SETUP	StringRange
config.ssw.15	TEXT	SAVE SETUP	StringRange
config.ssw.16	TEXT	SAVE SETUP	StringRange
gotoA	TEXT		StringRange
gotoB	TEXT		StringRange
progressReport.1	TEXT	R/O	StringRange R/O
progressIcon.1	TEXT	R/O	StringRange R/O
enabled.1	BOOLEAN	R/O	BooleanRange R/O (true false)
location.1	TEXT	R/O	StringRange R/O
nameA.1	TEXT	R/O	StringRange R/O
descriptionA.1	TEXT	R/O	StringRange R/O
nameB.1	TEXT	R/O	StringRange R/O
descriptionB.1	TEXT	R/O	StringRange R/O
comment.1	TEXT	R/O	StringRange R/O
progressReport.2	TEXT	R/O	StringRange R/O
progressIcon.2	TEXT	R/O	StringRange R/O
enabled.2	BOOLEAN	R/O	BooleanRange R/O (true false)
location.2	TEXT	R/O	StringRange R/O
nameA.2	TEXT	R/O	StringRange R/O
descriptionA.2	TEXT	R/O	StringRange R/O
nameB.2	TEXT	R/O	StringRange R/O
descriptionB.2	TEXT	R/O	StringRange R/O
comment.2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
progressReport.3	TEXT	R/O	StringRange R/O
progressIcon.3	TEXT	R/O	StringRange R/O
enabled.3	BOOLEAN	R/O	BooleanRange R/O (true false)
location.3	TEXT	R/O	StringRange R/O
nameA.3	TEXT	R/O	StringRange R/O
descriptionA.3	TEXT	R/O	StringRange R/O
nameB.3	TEXT	R/O	StringRange R/O
descriptionB.3	TEXT	R/O	StringRange R/O
comment.3	TEXT	R/O	StringRange R/O
progressReport.4	TEXT	R/O	StringRange R/O
progressIcon.4	TEXT	R/O	StringRange R/O
enabled.4	BOOLEAN	R/O	BooleanRange R/O (true false)
location.4	TEXT	R/O	StringRange R/O
nameA.4	TEXT	R/O	StringRange R/O
descriptionA.4	TEXT	R/O	StringRange R/O
nameB.4	TEXT	R/O	StringRange R/O
descriptionB.4	TEXT	R/O	StringRange R/O
comment.4	TEXT	R/O	StringRange R/O
progressReport.5	TEXT	R/O	StringRange R/O
progressIcon.5	TEXT	R/O	StringRange R/O
enabled.5	BOOLEAN	R/O	BooleanRange R/O (true false)
location.5	TEXT	R/O	StringRange R/O
nameA.5	TEXT	R/O	StringRange R/O
descriptionA.5	TEXT	R/O	StringRange R/O
nameB.5	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
descriptionB.5	TEXT	R/O	StringRange R/O
comment.5	TEXT	R/O	StringRange R/O
progressReport.6	TEXT	R/O	StringRange R/O
progressIcon.6	TEXT	R/O	StringRange R/O
enabled.6	BOOLEAN	R/O	BooleanRange R/O (true false)
location.6	TEXT	R/O	StringRange R/O
nameA.6	TEXT	R/O	StringRange R/O
descriptionA.6	TEXT	R/O	StringRange R/O
nameB.6	TEXT	R/O	StringRange R/O
descriptionB.6	TEXT	R/O	StringRange R/O
comment.6	TEXT	R/O	StringRange R/O
progressReport.7	TEXT	R/O	StringRange R/O
progressIcon.7	TEXT	R/O	StringRange R/O
enabled.7	BOOLEAN	R/O	BooleanRange R/O (true false)
location.7	TEXT	R/O	StringRange R/O
nameA.7	TEXT	R/O	StringRange R/O
descriptionA.7	TEXT	R/O	StringRange R/O
nameB.7	TEXT	R/O	StringRange R/O
descriptionB.7	TEXT	R/O	StringRange R/O
comment.7	TEXT	R/O	StringRange R/O
progressReport.8	TEXT	R/O	StringRange R/O
progressIcon.8	TEXT	R/O	StringRange R/O
enabled.8	BOOLEAN	R/O	BooleanRange R/O (true false)
location.8	TEXT	R/O	StringRange R/O
nameA.8	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
descriptionA.8	TEXT	R/O	StringRange R/O
nameB.8	TEXT	R/O	StringRange R/O
descriptionB.8	TEXT	R/O	StringRange R/O
comment.8	TEXT	R/O	StringRange R/O
progressReport.9	TEXT	R/O	StringRange R/O
progressIcon.9	TEXT	R/O	StringRange R/O
enabled.9	BOOLEAN	R/O	BooleanRange R/O (true false)
location.9	TEXT	R/O	StringRange R/O
nameA.9	TEXT	R/O	StringRange R/O
descriptionA.9	TEXT	R/O	StringRange R/O
nameB.9	TEXT	R/O	StringRange R/O
descriptionB.9	TEXT	R/O	StringRange R/O
comment.9	TEXT	R/O	StringRange R/O
progressReport.10	TEXT	R/O	StringRange R/O
progressIcon.10	TEXT	R/O	StringRange R/O
enabled.10	BOOLEAN	R/O	BooleanRange R/O (true false)
location.10	TEXT	R/O	StringRange R/O
nameA.10	TEXT	R/O	StringRange R/O
descriptionA.10	TEXT	R/O	StringRange R/O
nameB.10	TEXT	R/O	StringRange R/O
descriptionB.10	TEXT	R/O	StringRange R/O
comment.10	TEXT	R/O	StringRange R/O
progressReport.11	TEXT	R/O	StringRange R/O
progressIcon.11	TEXT	R/O	StringRange R/O
enabled.11	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
location.11	TEXT	R/O	StringRange R/O
nameA.11	TEXT	R/O	StringRange R/O
descriptionA.11	TEXT	R/O	StringRange R/O
nameB.11	TEXT	R/O	StringRange R/O
descriptionB.11	TEXT	R/O	StringRange R/O
comment.11	TEXT	R/O	StringRange R/O
progressReport.12	TEXT	R/O	StringRange R/O
progressIcon.12	TEXT	R/O	StringRange R/O
enabled.12	BOOLEAN	R/O	BooleanRange R/O (true false)
location.12	TEXT	R/O	StringRange R/O
nameA.12	TEXT	R/O	StringRange R/O
descriptionA.12	TEXT	R/O	StringRange R/O
nameB.12	TEXT	R/O	StringRange R/O
descriptionB.12	TEXT	R/O	StringRange R/O
comment.12	TEXT	R/O	StringRange R/O
progressReport.13	TEXT	R/O	StringRange R/O
progressIcon.13	TEXT	R/O	StringRange R/O
enabled.13	BOOLEAN	R/O	BooleanRange R/O (true false)
location.13	TEXT	R/O	StringRange R/O
nameA.13	TEXT	R/O	StringRange R/O
descriptionA.13	TEXT	R/O	StringRange R/O
nameB.13	TEXT	R/O	StringRange R/O
descriptionB.13	TEXT	R/O	StringRange R/O
comment.13	TEXT	R/O	StringRange R/O
progressReport.14	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
progressIcon.14	TEXT	R/O	StringRange R/O
enabled.14	BOOLEAN	R/O	BooleanRange R/O (true false)
location.14	TEXT	R/O	StringRange R/O
nameA.14	TEXT	R/O	StringRange R/O
descriptionA.14	TEXT	R/O	StringRange R/O
nameB.14	TEXT	R/O	StringRange R/O
descriptionB.14	TEXT	R/O	StringRange R/O
comment.14	TEXT	R/O	StringRange R/O
progressReport.15	TEXT	R/O	StringRange R/O
progressIcon.15	TEXT	R/O	StringRange R/O
enabled.15	BOOLEAN	R/O	BooleanRange R/O (true false)
location.15	TEXT	R/O	StringRange R/O
nameA.15	TEXT	R/O	StringRange R/O
descriptionA.15	TEXT	R/O	StringRange R/O
nameB.15	TEXT	R/O	StringRange R/O
descriptionB.15	TEXT	R/O	StringRange R/O
comment.15	TEXT	R/O	StringRange R/O
progressReport.16	TEXT	R/O	StringRange R/O
progressIcon.16	TEXT	R/O	StringRange R/O
enabled.16	BOOLEAN	R/O	BooleanRange R/O (true false)
location.16	TEXT	R/O	StringRange R/O
nameA.16	TEXT	R/O	StringRange R/O
descriptionA.16	TEXT	R/O	StringRange R/O
nameB.16	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
descriptionB.16	TEXT	R/O	StringRange R/O
comment.16	TEXT	R/O	StringRange R/O

9.388 N-Way-Switch-Latching

This page describes the device driver and the device window for the*N-Way-Switch-Latching* logical device. This device controls a latching n-way switch which is controlled by a dedicated drive circuit for each input. The switch is actuated by applying pulses of a defined length to the drive circuits. The switch position is determines by a set of position indication circuits read through PLC inputs.

The *N-Way-Switch* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

The n-way switch must be wired in a way that the drive coils of the switch are connected to consecutive output ports which appear in the same output register variable of the PLC driver. The same applies to the position indicating inputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets and shows the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	*description*
firstDrivePort	The address of the PLC port which activates the lowest switch position. The output ports for the subsequent switch positions are evaluated automatically. See the paragraph 'Port addressing' below for details.

parameter	***description***
firstIndicationPort	The address of the PLC port which indicated lowest switch position. The input ports for the subsequent switch positions are evaluated automatically. See the paragraph 'Port addressing' below for details.
resetPort	The address of a reset port to be activated in advance to each switch activation. If the switch does not require a reset pulse, leave this field empty. If 'resetPort' contains a port address (see 'Port addressing' below for a specification of PLC port addresses), the device first activates the reset port for one pulse width in order to reset the switch before it activates a switch port.
pulseWidth	The activation pulse width (msec) to be sent to the switch for a position change. When setting this parameter, you should be aware that pulses which are shorter than the IDLE time of the PLC driver's thread may get lost. If in doubt, set the pulse width to 1100 msec or in the VLC interface/device setup screen tune the IDLE time of the PLC driver thread to e.g. 200 msec.
positions	By default the device reports the state of the switch as '1' ... '4'. You may specify arbitrary texts for the switch positions. 'positions' contains a list of descriptions, separated by comma characters. The number of list elements also sets the number of output ports, the N-Way-Switch-Latching uses.
commFaultId	By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault.
The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).	

Fault flags

fault flag	***fault name***	***description***
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Position Indication	This fault is set if the switch position cannot be determined unambiguously. This is if not exactly one of the indication ports is 'high'.
faults.02	Wrong Position	This fault is set if the indicated switch position does not match the commanded one. The fault remains active until either the switch position is changed or a new position is commanded.
faults.03	Configuration Setup	This fault signals that the 'firstDrivePort' or the 'firstIndicationPort' parameter is not properly configured. The device expects, that these variables are set to values which match the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the **+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address ' *FANUC.Q00065+1* ' refers to the *2^1* bit in the variable ' *Q00065* ' provided by the ' *FANUC* ' device. In fact this addresses the output table register '%Q00066' at the PLC. Other PLC drivers may use a different naming scheme for the variables, but the concept always is the same.

In case of the N-Way-Switch-Latching device, the configured port address specifies the first of the (totally *n*) output ports which drive the switch. Configuration example:

```
firstDrivePort = PLC.Q00001+0
firstIndicationPort = PLC.I00001+0
positions = 1,2,3,4
```

This example configures a 4-way switch. It uses the output ports '*PLC.Q00001+0* ' .. ' *PLC.Q00001+3* ' to drive the switch. The input ports '*PLC.I00001+0* ' .. ' *PLC.I00001+3* ' indicate the switch position.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.firstDrivePort	TEXT	SAVE SETUP	StringRange
config.firstIndicationPort	TEXT	SAVE SETUP	StringRange
config.resetPort	TEXT	SAVE SETUP	StringRange
config.pulseWidth	INTEGER	SAVE SETUP	IntegerRange (200 .. 5000)
config.positions	TEXT	SAVE SETUP	StringRange
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (1 2 3 4)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Position Indication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Wrong Position)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.389 N-Way-Switch

This page describes the device driver and the device window for the*N-Way-Switch* logical device. This device controls a non-latching n-way switch which is controlled by a dedicated drive circuit for each input.

The *N-Way-Switch* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

The n-way switch must be wired in a way that the drive coils of the switch are connected to consecutive output ports which appear in the same output register variable of the PLC driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets end shows the (translated) PLC port state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
firstDrivePort	The address of the PLC which activates lowest switch position. The output ports for the subsequent switch positions are evaluated automatically. See the paragraph 'Port addressing' below for details.
includeOff	If you set this parameter to NO, the first selection from the 'positions' list activates the output port addressed by 'firstDrivePort'. With includeOff = YES, the first selection from the 'positions' list switches off all drive outputs. In this case the second selection activates the first output port. monitor.
positions	By default the device reports the state of the switch as '0' ... '4'. You may specify arbitrary texts for the switch positions. 'positions' contains a list of descriptions, separated by comma characters. The number of list elements also set the number of output ports, the N-Way-Switch uses.

parameter	*description*
commFaultId	By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault. The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).

Fault flags

fault flag	*fault name*	*description*
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Drive Combination	This fault is set if more than one output port is activated, or - with includeOff=NO - not port at all is set.
faults.02	Configuration Setup	This fault signals that the 'firstDrivePort' parameter ist not properly configured. The device expects, that 'drivePort' is set to a value which matches the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the**+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address ' *FANUC.Q00065+1* ' refers to the *2^1* bit in the variable ' *Q00065* ' provided by the ' *FANUC* ' device. In fact this addresses the output table register '%Q00066' at

the PLC. Other PLC drivers may use a different naming scheme for the variables, but the concept always is the same.

In case of the N-Way-Switch device, the configured port address specifies the first of the (totally *n*) output ports which drive the switch. Configuration example:

```
firstDrivePort = PLC.Q00001+0
includeOff = YES
positions = 0,1,2,3,4
```

This example configures a 4-way switch with an additional 'all-off' setting. It uses the output ports ' *PLC.Q00001+0* ' .. ' *PLC.Q00001+3* '. The switch position ' *0* ' clears all four output signals, switch position ' *3* ' for example activates the output port '*PLC.Q00001+2* '.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.firstDrivePort	TEXT	SAVE SETUP	StringRange
config.includeOff	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.positions	TEXT	SAVE SETUP	StringRange
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (0 1 2 3 4)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Drive Combination)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.390 NAIS-FP-8-Port-Switch

This page describes the device driver and the device window for the*NAIS-FP-8-Port-Switch* device. This device controls 8 single output ports of a NAiS/Matsushita PLC.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
port[1..8]	The address of the eight output ports to control.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The *NAIS-FP-8-Port-Switch* device reads back the port state it commands. On startup it

restores the last known configuration of the PLC.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.port1	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port2	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port3	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port4	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port5	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port6	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port7	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.port8	INTEGER	R/O	IntegerRange R/O (0 .. 2)
config.port1	TEXT	SAVE SETUP	StringRange
config.port2	TEXT	SAVE SETUP	StringRange
config.port3	TEXT	SAVE SETUP	StringRange
config.port4	TEXT	SAVE SETUP	StringRange
config.port5	TEXT	SAVE SETUP	StringRange
config.port6	TEXT	SAVE SETUP	StringRange
config.port7	TEXT	SAVE SETUP	StringRange
config.port8	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
position	INTEGER	SAVE	IntegerRange (0 .. 9)

9.391 NAIS-FP-AD-Test

The NAIS-FP-AD-Test driver accesses the data read by an A/D converter module connected to a FP0 programmable device. It is made for test purposes only and not recommended for general usage.

There is no further online help for this type of device. Please refer to the general description of device windows instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.value-0	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.value-2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.value-4	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.value-6	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.wx2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.wx3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.392 NAIS-FP-Alarm-Circuit

This page describes the device driver and the device window for the*NAIS-FP-Alarm-Circuit* device. This device monitors an alarm contact through an input port of a NAiS/Matsushita FPx programmable logic controller. The *NAIS-FP-Alarm-Circuit* device raises an alarm if the contact is opened, i.e. no current flows through the PLC input:

circuit	***logical state***	***alarm state***
OPEN	LOW	ALARM
CLOSED	HIGH	OK

There also is a NAIS-FP-Inv-Alarm-Circuit device available, which provides an inverted sense.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page is empty. It is contained for compatibility to other switch/monitoring devices of the *NAIS-FP-** family.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
port	The address of the input port to be monitored.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The fault message written by this device to the event log is named "XXXXXX Circuit FAULT" with XXXXXX being the name of the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Circuit)

9.393 NAIS-FP-Inv-Alarm-Circuit

This page describes the device driver and the device window for the*NAIS-FP-Inv-Alarm-Circuit* device. This device monitors an alarm contact through an input port of a NAiS/Matsushita FPx programmable logic controller. The *NAIS-FP-Inv-Alarm-Circuit* device raises an alarm if the contact is closed, i.e. current flows through the PLC input:

circuit	***logical state***	***alarm state***
OPEN	LOW	OK

circuit	*logical state*	*alarm state*
CLOSED	HIGH	ALARM

There also is a NAIS-FP-Alarm-Circuit device available, which provides an inverted sense.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page is empty. It is contained for compatibility to other switch/monitoring devices of the *NAIS-FP-** family.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	*description*
address	The packet address of the PLC device. Usually this is "01".
port	The address of the input port to be monitored.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.

- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The fault message written by this device to the event log is named "XXXXXX Circuit FAULT" with XXXXXX being the name of the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Circuit)

9.394 NAIS-FP-Inv-Pulse-Output

This page describes the device driver and the device window for the*NAIS-FP-Inv-Pulse-Output* device. This device controls one single output port of a NAiS/Matsushita PLC. The output ist deactivated for a defined time everytime when an arbitrary position command arrives at the device.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Commands the switch position. In a task-oriented user interface the NAIS-FP-Pulse-Output usually will be bound to a button which starts a pule at the output when pressed. Here, with the standard user interface for switchtes, you have to enter an arbitrary value to the 'position' text field.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
port	The address of the output port to control.
pulse	The pulse duration (seconds). Be aware, that during the pulse time all polling of inputs at this PLC device is paused!
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The *NAIS-FP-Pulse-Output* device does not read back a switch state. The output is assumed to normally deactivated.

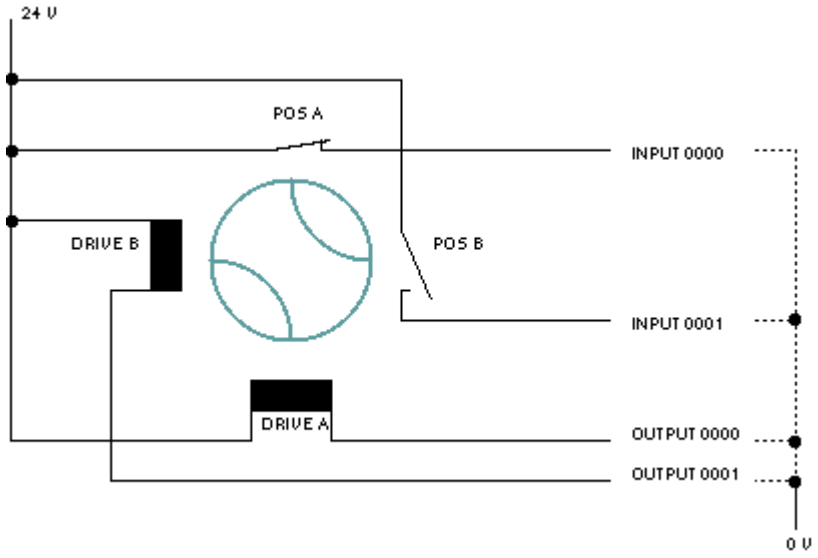
Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
config.pulse	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 60.0)
position	TEXT	NOPRESET	StringRange

9.395 NAIS-FP-Latching-Relay

This page describes the device driver and the device window for the*NAIS-FP-Latching-Relay* device. This device controls a latching relay or wave guide switch. To change the position, it applies a current pulse to one of the motors/drive coils at the switch. On the other hand it monitors the switch position by means of two position contacts. The figure below shows an example how to connect a wave guide switch to the PLC.



Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and

set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
drivePortA	The address of the output port connected to the drive A motor/coil.
drivePortB	The address of the output port connected to the drive B motor/coil.
readbackPortA	The address of the input port connected to the contact closed at position A.
readbackPortB	The address of the input port connected to the contact closed at position B.
pulse	The duration of the current impulse sent in order to change the position.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is

loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.

- The A/B switch position assignment easily can be reverted by exchanging the port numbers at the configuration parameters.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.drivePortA	TEXT	SAVE SETUP	StringRange
config.drivePortB	TEXT	SAVE SETUP	StringRange
config.readbackPortA	TEXT	SAVE SETUP	StringRange
config.readbackPortB	TEXT	SAVE SETUP	StringRange
config.pulse	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 2.0)
position	CHOICE		EnumRange (A B)
internal.readA	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.readB	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Indifferent position)

9.396 NAIS-FP-Protection-Switch

This page describes the device driver and the device window for the*NAIS-FP-Protection-Switch* device. This device controls a NAiS/Matsushita PLC running a program that implements a 1:1 protection switch. The PLC recognizes several device alarms and controls a waveguide switch. The driver exclusively works with a PLC programmed in this special way.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the global Enable/Disable switch for the protection.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
operationMode	Defines if the protection shall be disabled after it has been triggered once.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not work in "PGM" mode.
- The wave guide switch is operated at the I/O terminals 0000/0001 of the PLC. To control the switch position directly you may define a NAIS-FP-Latching-Relay device on these ports.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.operationMode	CHOICE	SETUP	EnumRange (SW-ONCE SW-ALWAYS)
position	CHOICE		EnumRange (ENABLED DISABLED)
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (WG switch)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FP0 program)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Has switched)

9.397 NAIS-FP-Pulse-Output

This page describes the device driver and the device window for the*NAIS-FP-Pulse-Output* device. This device controls one single output port of a NAiS/Matsushita PLC. The output ist activated for a defined time everytime when an arbitrary position command arrives at the device.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Commands the switch position. In a task-oriented user interface the NAIS-FP-Pulse-Output usually will be bound to a button which starts a pule at the output when pressed.

Here, with the standard user interface for switchtes, you have to enter an arbitrary value to the 'position' text field.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
port	The address of the output port to control.
pulse	The pulse duration (seconds). Be aware, that during the pulse time all polling of inputs at this PLC device is paused!
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The *NAIS-FP-Pulse-Output* device does not read back a switch state. The output is assumed to normally deactivated.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
config.pulse	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 2.0)
position	TEXT	NOPRESET	StringRange

9.398 NAIS-FP-RTD-Module

This page describes the device driver and the device window for the*NAIS-FP-RTD-Module* device. This module provides a six-channel PT100/PT1000 temperature sensor interface for the FP0 PLC. Up to three RTD modules may be used with a single FP0 PLC.

For each channel the driver provides individual low/high limits and fault flags which indicate if the temperature value is outside the limits. By default, the temperature sensors are named 'CH-0' .. CH-5' like the RTD module documentation does. The name can be customized, which also changes the name of the fault flags belonging to this channel.

Although 'PT' type temperature sensors are quite accurate without any additional calibration, the device driver provides an individual calibration offset setting for each channel which can be set at the maintenance / setup page.

The *NAIS-FP-RTD-Module* device may be used together with other NAIS-FP-* devices referring to the same FP0 PLC. See paragraph 'Configuration hints' at the and of this chapter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual temperature values and lets you set individual fault limits.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
tempSensor	The position (1..3) where the RTD module is located. 1 means directly beside the PLC CPU module, 2 means with one module in between and so on.
ch0.enable	
...	
ch5.enable	With this switch you enable / disable the processing of each particular channel. Channels set to "NO" (disabled) always read *0.0°C* and show no limit faults, regardless of the threshold settings. In the device window, disabled channels appear greyed. By disabling unused channels you can prevent them from showing wrong values if no sensor is connected.
ch0.offset	
...	
ch5.offset	The offsets defined here are added to the raw measurement value before the value is displayed or compared against the thresholds. Use this if you need to calibrate the temperature display.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Defining limits

At the first page of the device window, a low and a high limit value can be set. If the temperature value exceeds one of the limits, a fault is generated and a message is written to the log. To disable the limit checking, set both limit values to zero.

Configuration hints

1. You must set the configuration switches at the RTD module for the type of sensors you use (see the documentation coming with the module).
2. If you want to mix regular I/O modules and a temperature module at one FP0 CPU, you must install the RTD module most far away from the CPU.
3. If you want to use the NAIS-FP0 device together with this driver, set the 'extModules'

configuration parameter at the NAIS-FP0 device that the temperature module gets not polled by it.Example:You have an FP0 CPU with one input expansion module and one RTD module. Attach the input expansion to the CPU, the RTD module to the expansion. At the device setup, define the interface to the FP0, choose the NAIS-FPx protocol for it. Attach a NAIS-FP0 device for the I/O ports and a NAIS-FP-RTD-Module device for the measurement module.At the NAIS-FP0 configuration setup set the 'extModules' parameter to '1', at the NAIS-FP-RTD-Module configuration setup set the 'tempModule' parameter to '2'.

Refresh intervals

Due to the principle the of operation of the communication interface between the sat-nms system and the RTD module, there is no fixed refresh cycle time for a particular measurement channel. The driver polls the FP0 every 4 seconds. With each poll, the FP0 replies two of the six channels, in random order.

At average, each channel gets updated every 12 seconds. Due to the random sequence at which the channels get updated, it may happen that a channel gets not updated for a longer time.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.rawval	TEXT	R/O	StringRange R/O
config.tempModule	CHOICE	SAVE SETUP	EnumRange (1 2 3)
config.ch0.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.ch1.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.ch2.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.ch3.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
config.ch4.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.ch5.enable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.ch0.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.ch1.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.ch2.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.ch3.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.ch4.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.ch5.offset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
internal.initModule	TEXT	R/O	StringRange R/O
ch0.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch0.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch0.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch0.name	TEXT	SAVE	StringRange
ch1.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch1.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch1.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch1.name	TEXT	SAVE	StringRange
ch2.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch2.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch2.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch2.name	TEXT	SAVE	StringRange
ch3.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch3.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch3.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch3.name	TEXT	SAVE	StringRange
ch4.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch4.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
ch4.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch4.name	TEXT	SAVE	StringRange
ch5.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ch5.max	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch5.min	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
ch5.name	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (CH-0 low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (CH-0 high limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CH-1 low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (CH-1 high limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (CH-2 low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (CH-2 high limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (CH-3 low limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (CH-3 high limit)
faults.09	ALARM	R/O	AlarmFlagRange R/O (CH-4 low limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (CH-4 high limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (CH-5 low limit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (CH-5 high limit)

9.399 NAIS-FP-Relay

This page describes the device driver and the device window for the*NAIS-FP-Relay* device. This device controls a non-latching relay / coax switch with position contacts. The non-activated position of the relay is called "A", the energized position is called "B".

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
drivePort	The address of the output port connected to the relay coil.
readbackPortA	The address of the input port connected to the contact closed at position A.
readbackPortB	The address of the input port connected to the contact closed at position B.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.drivePort	TEXT	SAVE SETUP	StringRange
config.readbackPortA	TEXT	SAVE SETUP	StringRange
config.readbackPortB	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (A B)
internal.readA	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.readB	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Indifferent position)

9.400 NAIS-FP-Simple-Input

This page describes the device driver and the device window for the*NAIS-FP-Simple-Input* device. This device monitors one single input port of a NAiS/Matsushita PLC and reports the port state to the user.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
port	The address of the input port to monitor.
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
position	CHOICE	R/O	EnumRange R/O (0 1)

9.401 NAIS-FP-Simple-Output

This page describes the device driver and the device window for the*NAIS-FP-Simple-Output* device. This device controls one single output port of a NAiS/Matsushita PLC.

Multiple devices of this type (and other members of the*NAIS-FP-** -family) may refer to one single PLC device. To do so, configure these devices to share the serial interface of the PLC and set the configuration parameters of each particular switch/monitoring device to the same PLC (address) but to individual ports.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	*description*
address	The packet address of the PLC device. Usually this is "01".
port	The address of the output port to control.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- Port addresses used with the PLC are 4 digit hex numbers (upper case digits 'A'..'F', always padded to 4 characters with leading zeroes). The ports on a FPx device itself start at "0000" (inputs and output use their own address spaces). See the PLC manual for the addresses of I/O ports located at extension modules.
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.
- The *NAIS-FP-Simple-Output* device reads back the port state it commands. On startup it reads the actual port state from the PLC.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.port	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (OFF ON)

9.402 NAIS-FP-TxOn-Switch

This page describes the device driver and the device window for the*NAIS-FP-TxOn-Switch* device. This device controls a NAiS/Matsushita PLC running a program that switches the tx switch if two alarms (e.g. from satellite modems) are active longer than 15 minutes. The PLC recognizes two device alarms and controls one waveguide switch. The driver exclusively works with a PLC programmed in this special way.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the global Enable/Disable switch for the protection.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not work in "PGM" mode.
- The wave guide switch is operated at the I/O terminals 0002/0003 of the PLC. To control the switch position directly you may define a NAIS-FP-Latching-Relay device on these ports.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
position	CHOICE		EnumRange (ENABLED DISABLED)
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Has switched)

9.403 NAIS-FP-Watchdog-Output

This page describes the device driver and the device window for the*NAIS-FP-Watchdog-Output* device. This device sends a signal to a NAiS/Matsushita PLC running a program that implements a watchdog which closes a circuit if the M&C computer stops sending signals to the PLC. The driver exclusively works with a PLC programmed in this special way.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Has no function withthis driver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	*description*
address	The packet address of the PLC device. Usually this is "01".

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* . With multiple *NAIS-FP-** -devices referring to the same PLC, the line speed needs to be setup only once.

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not work in "PGM" mode.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
position	CHOICE	R/O	EnumRange R/O (ON)

9.404 NAIS-FP0-REG

This page describes the device driver to read and write up to 16 registers of an FP0 Module.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Inputs and outputs
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
module	Select the module
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* .

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.module	CHOICE	SAVE SETUP	EnumRange (CPU EXT1 EXT2 EXT3)
reg.00	BOOLEAN		BooleanRange (true false)
reg.01	BOOLEAN		BooleanRange (true false)
reg.02	BOOLEAN		BooleanRange (true false)

name	*type*	*flags*	*range*
reg.03	BOOLEAN		BooleanRange (true false)
reg.04	BOOLEAN		BooleanRange (true false)
reg.05	BOOLEAN		BooleanRange (true false)
reg.06	BOOLEAN		BooleanRange (true false)
reg.07	BOOLEAN		BooleanRange (true false)
reg.08	BOOLEAN		BooleanRange (true false)
reg.09	BOOLEAN		BooleanRange (true false)
reg.10	BOOLEAN		BooleanRange (true false)
reg.11	BOOLEAN		BooleanRange (true false)
reg.12	BOOLEAN		BooleanRange (true false)
reg.13	BOOLEAN		BooleanRange (true false)
reg.14	BOOLEAN		BooleanRange (true false)
reg.15	BOOLEAN		BooleanRange (true false)
registers	HEX	R/O	HexRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
internal.lo	HEX	R/O	HexRange R/O (0 .. 0)
internal.hi	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt	HEX	R/O	HexRange R/O (0 .. 0)
internal.address	TEXT	R/O	StringRange R/O
internal.label	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Self diagnostics)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fuse (output module))
faults.03	ALARM	R/O	AlarmFlagRange R/O (Special module)
faults.04	ALARM	R/O	AlarmFlagRange R/O (I/O Compare)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Battery)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Program flow)

9.405 NAIS-FP0-RO

This page describes the device driver and the device window for the a read only driver to monitor up to 16 inputs and outputs of an FP0 Module

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Inputs and outputs
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".
module	Select the module
Serial	Remember to set the line parameters for the serial interface connected to the FPx PLC device. It requires the somewhat unusual setting *9600,O,8,1* .

Remarks

- The device address of a FPx PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.module	CHOICE	SAVE SETUP	EnumRange (CPU EXT1 EXT2 EXT3)
in.00	BOOLEAN	R/O	BooleanRange R/O (true false)
in.01	BOOLEAN	R/O	BooleanRange R/O (true false)
in.02	BOOLEAN	R/O	BooleanRange R/O (true false)
in.03	BOOLEAN	R/O	BooleanRange R/O (true false)
in.04	BOOLEAN	R/O	BooleanRange R/O (true false)
in.05	BOOLEAN	R/O	BooleanRange R/O (true false)
in.06	BOOLEAN	R/O	BooleanRange R/O (true false)
in.07	BOOLEAN	R/O	BooleanRange R/O (true false)
in.08	BOOLEAN	R/O	BooleanRange R/O (true false)
in.09	BOOLEAN	R/O	BooleanRange R/O (true false)
in.10	BOOLEAN	R/O	BooleanRange R/O (true false)
in.11	BOOLEAN	R/O	BooleanRange R/O (true false)
in.12	BOOLEAN	R/O	BooleanRange R/O (true false)
in.13	BOOLEAN	R/O	BooleanRange R/O (true false)
in.14	BOOLEAN	R/O	BooleanRange R/O (true false)
in.15	BOOLEAN	R/O	BooleanRange R/O (true false)
out.00	BOOLEAN	R/O	BooleanRange R/O (true false)
out.01	BOOLEAN	R/O	BooleanRange R/O (true false)
out.02	BOOLEAN	R/O	BooleanRange R/O (true false)
out.03	BOOLEAN	R/O	BooleanRange R/O (true false)
out.04	BOOLEAN	R/O	BooleanRange R/O (true false)
out.05	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
out.06	BOOLEAN	R/O	BooleanRange R/O (true false)
out.07	BOOLEAN	R/O	BooleanRange R/O (true false)
out.08	BOOLEAN	R/O	BooleanRange R/O (true false)
out.09	BOOLEAN	R/O	BooleanRange R/O (true false)
out.10	BOOLEAN	R/O	BooleanRange R/O (true false)
out.11	BOOLEAN	R/O	BooleanRange R/O (true false)
out.12	BOOLEAN	R/O	BooleanRange R/O (true false)
out.13	BOOLEAN	R/O	BooleanRange R/O (true false)
out.14	BOOLEAN	R/O	BooleanRange R/O (true false)
out.15	BOOLEAN	R/O	BooleanRange R/O (true false)
inputs	HEX	R/O	HexRange R/O (0 .. 0)
outputs	HEX	R/O	HexRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
internal.lo	HEX	R/O	HexRange R/O (0 .. 0)
internal.hi	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt	HEX	R/O	HexRange R/O (0 .. 0)
internal.address	TEXT	R/O	StringRange R/O
internal.label	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Self diagnostics)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fuse (output module))
faults.03	ALARM	R/O	AlarmFlagRange R/O (Special module)
faults.04	ALARM	R/O	AlarmFlagRange R/O (I/O Compare)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Battery)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Program flow)

9.406 NAIS-FP0

This page describes the device driver and the device window for the*NAIS-FP0* device. This device controls the I/O ports of a Matsushita / NAiS FP0 PLC. It can handle the ports at the PLC base unit and at up to three extension moules. It communicates to the PLC through the programming port at the PLC base unit. (see the paragraph at the end of this chapter).

The *NAIS-FP0* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*NAIS-FP0* device driver directly. There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *NAIS-FP0* device driver as a low level driver to access the physical I/O ports.

Port addressing

The device driver gives access to the I/O ports through four integer variables each for inputs and for outputs. Each variable stands for up to 32 input or output ports. The variables are mapped to the PLC's internal I/O addresses following the NAiS addressing conventions:

The variables 'X0000' and 'Y0000' refer to the ports at the PLC base unit, 'X0020' and 'Y0020' to the ports at the first extenstion motule etc. Within the variables, the IO ports are addressed by bit positions. Example:

An input port which appears in the FP0 PLC as 'X0006' is accessed through the variable X0000 of the device driver. It is represented by the bit 2^6 in the variable (the least significant bit represents the 'X0000' input).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Fanuc PLC.

parameter	***description***
address	The packet address of the PLC device. Usually this is "01".

parameter	*description*
extModules	Select the number of extension modules you have connected to the FP0. The driver ignores the changes to 'Y00x0' variables of unused modules and it does not read out / update the corresponding 'X00x0' variables.
Serial	Remember to set the line parameters for the serial interface connected to the NAiS FP0 PLC. The setting required by the FP0 is *9600,O,8,1* .

Remarks

- The device address of a FP0 PLC usually is "01" (two digit, the leading zero is mandatory).
- The FPx PLC must be switched to "RUN" mode. It will not reply in "PGM" mode.
- The M&C software uses the PLC as a "dumb" I/O device. It assumes that no program is loaded to the PLC (you can use it "out of the box"). If you need to run a program on the same PLC device anyhow, ensure that the program does not access the same I/O registers the M&C addresses.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.dummy	HEX	R/O	HexRange R/O (0 .. 0)
config.extModules	CHOICE	SAVE SETUP	EnumRange (0 1 2 3)
X0000	HEX	R/O	HexRange R/O (0 .. 0)
X0020	HEX	R/O	HexRange R/O (0 .. 0)
X0040	HEX	R/O	HexRange R/O (0 .. 0)
X0060	HEX	R/O	HexRange R/O (0 .. 0)
Y0000	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
Y0020	HEX		HexRange (0 .. 0)
Y0040	HEX		HexRange (0 .. 0)
Y0060	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

9.407 NDSatCom-1Axis-ACU

This page describes the device driver and the device window for the ND SatCom antenna control unit. The NDSatCom-1Axis-ACU driver directly controls outdoor units which implement the three axes on three individual single PCBs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.The set point parameters are shown with a resolution of 0.1 deg for azimuth/elevation and 1 deg for the polarization. This is to make the spin buttons work in a convenient way. You still may enter azimuth/elevation values with a 0.01 deg resolution, the antenna controller accepts this.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	The address field **must** be empty for third type od ACU. The three boards of the ACU expect individual addresses for each axis. The driver contains hard coded the usual assignment '1' = azimuth, '2' = elevation and '3' = polarization.
azLoLimit	The low azimuth limit. You should set this value at least 2 degrees above the limit switch position.

parameter	*description*
azHiLimit	The high azimuth limit. You should set this value at least 2 degrees below the limit switch position.
elLoLimit	The low elevation limit. You should set this value at least 2 degrees above the limit switch position.
elHiLimit	The high elevation limit. You should set this value at least 2 degrees below the limit switch position.
plLoLimit	The low polarization limit. You should set this value at least 2 degrees above the limit switch position.
plHiLimit	The high polarization limit. You should set this value at least 2 degrees below the limit switch position.
azMotorInit	The motor parameter and angle calibration string required by the ACU (azimuth axis). **You must take this string from the NDSatCom antenna control software, the antenna may be damaged if the ACU is operated without or with wrong motor initialization data!**
elMotorInit	The motor parameter and angle calibration string required by the ACU (elevation axis). **You must take this string from the NDSatCom antenna control software, the antenna may be damaged if the ACU is operated without or with wrong motor initialization data!**
plMotorInit	The motor parameter and angle calibration string required by the ACU (polarization axis). **You must take this string from the NDSatCom antenna control software, the antenna may be damaged if the ACU is operated without or with wrong motor initialization data!**

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
internal.mx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.my	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.motorInit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 360.00)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 360.00)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 90.00)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 90.00)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
config.azMotorInit	TEXT	SAVE SETUP	StringRange
config.elMotorInit	TEXT	SAVE SETUP	StringRange
config.plMotorInit	TEXT	SAVE SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0 .. 0)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Limit)

9.408 NDSatCom-3Axis-ACU

This page describes the device driver and the device window for the ND SatCom antenna control unit. The NDSatCom-3Axis-ACU driver directly controls outdoor units which implement all three axes on one single PCB.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.The set point parameters are shown with a resolution of 0.1 deg for azimuth/elevation and 1 deg for the polarization. This to make the spin buttons work in a convenient way. You still may enter azimuth/elevation values with a 0.01 deg resolution, the antenna controller accepts this.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The ACU usually expects the address "1".
azLoLimit	The low azimuth limit. You should set this value at least 2 degrees above the limit switch position.
azHiLimit	The high azimuth limit. You should set this value at least 2 degrees below the limit switch position.
azCalibration	The resolver calibration offset.
elLoLimit	The low elevation limit. You should set this value at least 2 degrees above the limit switch position.
elHiLimit	The high elevation limit. You should set this value at least 2 degrees below the limit switch position.
elCalibration	The resolver calibration offset.

parameter	*description*
plLoLimit	The low polarization limit. You should set this value at least 2 degrees above the limit switch position.
plHiLimit	The high polarization limit. You should set this value at least 2 degrees below the limit switch position.
plCalibration	The resolver calibration offset.
motorInit	The motor parameter string needed by the ACU. The string begin and end with a comma. **You must take this string from the NDSatCom antenna control software, the antenna may be damaged if the ACU is operated without or with wrong motor initialization data!**

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.mx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.my	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mz	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ack	TEXT	R/O	StringRange R/O
internal.azUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.azDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.elUp	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.elDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.motorInit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.calInit	BOOLEAN	R/O	BooleanRange R/O (true false)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 360.00)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 360.00)
config.azCalibration	FLOAT	SAVE SETUP	DoubleRange (0.000000 .. 0.000000)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 90.00)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 90.00)
config.elCalibration	FLOAT	SAVE SETUP	DoubleRange (0.000000 .. 0.000000)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
config.plCalibration	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.motorInit	TEXT	SAVE SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0 .. 0)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL high limit)

9.409 NDSatCom-ACU4100A-SNG

This page describes the device driver and the device window for the ND SatCom ACU4100A-SNG for car mounted antennas.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing and provides controls to park / unpark the antenna.
- --- The 'Calibrate / Azimuth Scan' page provides controls to calibrate the antenna's azimuth reading by adjusting the antenna to aknown satellite.
- --- The 'State' page shown the actual readings reported by the antenna controller.
- --- The 'Auxiliary I/O' page gives access to the auxiliary relay outputs / optocoupler inputs of the ACU.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	The ACU usually expects the address "1".
Serial	Remember to set the line parameters for the serial interface connected to the ACU. *19200,N,8,1* are the settings the ACU expects.

Azimuth calibration

To calibrate the azimuth reading for the actual location of the car, the sat-nms device driver

supports you performing the following procedure:

1. Select a well known satellite (e.g. Hotbird 13° East) and calculate the azimuth / elevation angles for the geodetic position of the antenna.
2. Move the antenna up to the calculated elevation.
3. With a compass (or just by eye) estimate the azimuth direction where you expect the satellite.
4. Move the antenna to point some degrees more in the east.
5. With the scan feature, found at the 'Calibrate / Azimuth Scan' page, slowly move the antenna to the position where the satellite is expected.
6. Watch the spectrum analyzer or the IRD signal lock indicator. Press STOP as soon as the satellite has been found.
7. Fine-adjust the antenna for best receive quality.
8. Again at the 'Calibrate / Azimuth Scan' page, enter the calculated azimuth angle in the 'azimuth target' field an press 'CALIBRATE'.

The antenna's azimuth reading is now calibrated.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.temperature	INTEGER		IntegerRange (0 .. 0)
internal.m	HEX	R/O	HexRange R/O (0 .. 0)
internal.s	HEX	R/O	HexRange R/O (0 .. 0)
internal.b	HEX	R/O	HexRange R/O (0 .. 0)
internal.init	HEX	R/O	HexRange R/O (0 .. 0)
internal.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.azDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.elUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.elDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plDn	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0 .. 0)
azcal	FLOAT		DoubleRange (0.00 .. 0.00)
azoffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
calibrate	TEXT		StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.inclin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.analog1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.analog2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
aux.relay3	CHOICE		EnumRange (0 1)
aux.relay4	CHOICE		EnumRange (0 1)
aux.input1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.input2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.input3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.input4	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
scanSpeed	INTEGER	SAVE	IntegerRange (0 .. 100)
scanDest	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
reset	TEXT	NOPRESET	StringRange
park	TEXT	NOPRESET	StringRange
unpark	TEXT	NOPRESET	StringRange
scan	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Emergency STOP)

9.410 NDSatCom-ACU4100B

This page describes the device driver and the device window for the ND SatCom ACU4100B for fixed station antennas.

The driver was tested with the following FW releases:

- FW 3.51 03.11.2000
- FW 4.02 30.06.2004
- FW 5.00 16.09.2009

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing and provides controls to stop the antenna.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The ACU usually expects the address "1".
Serial	Remember to set the line parameters for the serial interface connected to the ACU. *19200,N,8,1* are the settings the ACU expects.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.m	HEX	R/O	HexRange R/O (0 .. 0)
internal.s	HEX	R/O	HexRange R/O (0 .. 0)
internal.init	HEX	R/O	HexRange R/O (0 .. 0)
internal.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azUp	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.azDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.elUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.elDn	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plUp	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.plDn	BOOLEAN	R/O	BooleanRange R/O (true false)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0 .. 0)
config.azoffs	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL high limit)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Emergency STOP)

9.411 NDSatCom-ACU5000-SNG

This page describes the device driver and the device window for the ND SatCom ACU5000 antenna controller for car mounted antennas.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main parameter for automatic pointing (APS) and stowing/unstowing the antenna
- --- This page contains the parameters for manual pointing / jogging the antenna.
- --- Displays the station location parameters like geodetic location and orientation angles.
- --- This page lets you edit the satellite list stored in the ACU.
- --- Contains the general parameters controlling the automatic pointing system.
- --- Contains the SKYWAN specific parameters.
- --- Contains the DVB specific parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- azJogStep --- The step delta used to jog the azimuth angle in manual mode.
- elJogStep --- The step delta used to jog the elevation angle in manual mode.
- plJogStep --- The step delta used to jog the polarization angle in manual mode.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.softwareType	TEXT	R/O	StringRange R/O
info.hasDvbAps	TEXT	R/O	StringRange R/O
info.hasSkywanAps	TEXT	R/O	StringRange R/O
info.skywanEnabled	TEXT	R/O	StringRange R/O
info.summaryStatus	TEXT	R/O	StringRange R/O
info.accessStatus	TEXT	R/O	StringRange R/O
info.manualControl	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.odu	TEXT	R/O	StringRange R/O
config.azJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plJogStep	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.initDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 180)
config.confirmLnbPatch	CHOICE	SAVE SETUP	EnumRange (AUTO MANUAL)
tmp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.stowedaz	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.stowedel	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.stowedpl	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.trueAz	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.trueEl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.stow	TEXT	R/O	StringRange R/O
state.stowed	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.gps	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.antRef	CHOICE	R/O	EnumRange R/O (NO YES)
state.satellite	TEXT	R/O	StringRange R/O
satellite	TEXT	R/O	StringRange R/O
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
park	TEXT	NOPRESET	StringRange
unpark	TEXT	NOPRESET	StringRange
scan	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
step	TEXT	NOPRESET	StringRange
station.height	FLOAT		DoubleRange (0.0 .. 0.0)
station.latitude	FLOAT		DoubleRange (0.00000 .. 0.00000)
station.longitude	FLOAT		DoubleRange (0.00000 .. 0.00000)
station.course	FLOAT		DoubleRange (0.000 .. 0.000)
station.nick	FLOAT		DoubleRange (0.000 .. 0.000)
station.roll	FLOAT		DoubleRange (0.000 .. 0.000)
station.mountOffset	FLOAT		DoubleRange (0.000 .. 0.000)
station.reference	CHOICE		EnumRange (EARTH MOUNT)
station.recalcSats	TEXT		StringRange
aps.state	TEXT	R/O	StringRange R/O
hasSetDest	BOOLEAN	R/O	BooleanRange R/O (true false)
aps.progress	INTEGER	R/O	IntegerRange R/O (0 .. 100)
aps.signal	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
aps.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
aps.lastError	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
aps.emergencyOff	TEXT	NOPRESET	StringRange
aps.off	TEXT	NOPRESET	StringRange
aps.pointing.start	TEXT	NOPRESET	StringRange
aps.pointing.stop	TEXT	NOPRESET	StringRange
aps.select.netconf	CHOICE		EnumRange (-1 UNKNOWN)
aps.requestLnbPatch	TEXT	NOPRESET	StringRange
aps.confirmLnbPatch	TEXT	NOPRESET	StringRange
aps.lnbPatch.01.in	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aps.lnbPatch.01.out	INTEGER	R/O	IntegerRange R/O (0 .. 0)
selectedNetworkType	TEXT	R/O	StringRange R/O
aps.scanOrientation	CHOICE		EnumRange (AUTO FULL FRONT LEFT RIGHT)
aps.azMaxSectors	INTEGER		IntegerRange (0 .. 0)
aps.azSearchRange	FLOAT		DoubleRange (0.000 .. 0.000)
aps.azScanIncrFactor	INTEGER		IntegerRange (0 .. 0)
aps.azScanRangeOffs	FLOAT		DoubleRange (0.000 .. 0.000)
aps.azLockHuntRange	FLOAT		DoubleRange (0.000 .. 0.000)
aps.azLockHuntStep	FLOAT		DoubleRange (0.000 .. 0.000)
aps.azScanOn	CHOICE		EnumRange (YES NO)
aps.elAntenna3dBSize	FLOAT		DoubleRange (0.00 .. 0.00)
aps.elMaxDeviation	FLOAT		DoubleRange (0.00 .. 0.00)
aps.elMaxNoOfIterations	INTEGER		IntegerRange (0 .. 0)
aps.elNoOfMeasurePoints	INTEGER		IntegerRange (0 .. 0)
aps.elSearchOn	CHOICE		EnumRange (YES NO)
aps.polMaxMeanSquare	FLOAT		DoubleRange (0.000000 .. 0.000000)
aps.polMaxRetries	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
aps.polStepSize	FLOAT		DoubleRange (0.000 .. 0.000)
aps.polMinNrOfPoints	INTEGER		IntegerRange (0 .. 0)
aps.polMaxDeviation	FLOAT		DoubleRange (0.000 .. 0.000)
aps.polStep3DeltaPsi	FLOAT		DoubleRange (0.00 .. 0.00)
aps.polUseOnError	CHOICE		EnumRange (YES NO)
aps.polTruncatePercent	INTEGER		IntegerRange (0 .. 0)
aps.polSearchOn	CHOICE		EnumRange (YES NO)
aps.useLocTable	CHOICE		EnumRange (YES NO)
aps.nickRollTolerance	FLOAT		DoubleRange (0.000 .. 0.000)
aps.polAngleTolerance	FLOAT		DoubleRange (0.000 .. 0.000)
aps.compassMeasPos	CHOICE		EnumRange (YES NO)
aps.compassMeasElevation	FLOAT		DoubleRange (0.000 .. 0.000)
skywan.networks	CHOICE		EnumRange (NONE)
skywan.netName	TEXT		StringRange
skywan.netDescr	TEXT		StringRange
skywan.netConfig	TEXT		StringRange
skywan.netSatellite	CHOICE		EnumRange (UNKNOWN)
skywan.netPol	CHOICE		EnumRange (HORIZONTAL VERTICAL LHCP RHCP)
si	INTEGER	R/O	IntegerRange R/O (0 .. 0)
skywan.add	TEXT	NOPRESET	StringRange
skywan.delete	TEXT	NOPRESET	StringRange
dvb.networks	CHOICE		EnumRange (NONE)
dvb.netName	TEXT		StringRange
dvb.netDescr	TEXT		StringRange
dvb.netLineSetting	CHOICE		EnumRange (UNKNOWN)

name	*type*	*flags*	*range*
dvb.netPol	CHOICE	R/O	EnumRange R/O (HORIZONTAL VERTICAL)
dvb.netUseTsId	CHOICE		EnumRange (YES NO)
dvb.netTsId	INTEGER		IntegerRange (0 .. 0)
di	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dvb.add	TEXT	NOPRESET	StringRange
dvb.delete	TEXT	NOPRESET	StringRange
satlist.index	INTEGER	R/O	IntegerRange R/O (0 .. 0)
satlist.names	CHOICE		EnumRange (NONE)
satlist.name	TEXT		StringRange
satlist.orbit	FLOAT		DoubleRange (0.000 .. 0.000)
satlist.polOffset	FLOAT		DoubleRange (0.000 .. 0.000)
satlist.beaconFreq	FLOAT		DoubleRange (0.000 .. 0.000)
satlist.polPlane	CHOICE		EnumRange (HORIZONTAL VERTICAL LHCP RHCP)
satlist.azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
satlist.elevation	FLOAT		DoubleRange (0.000 .. 0.000)
satlist.polarization	FLOAT		DoubleRange (0.000 .. 0.000)
ai	INTEGER	R/O	IntegerRange R/O (0 .. 0)
satlist.add	TEXT	NOPRESET	StringRange
satlist.delete	TEXT	NOPRESET	StringRange
satlist.move	TEXT	NOPRESET	StringRange
dvb.linesettings	CHOICE		EnumRange (UNKNOWN)
dvb.lsName	TEXT		StringRange
dvb.lsDescr	TEXT		StringRange
dvb.lsSatellite	CHOICE		EnumRange (UNKNOWN)
dvb.lsRxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
dvb.lsSymRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
dvb.lsFec	CHOICE		EnumRange (1/2 1/4 1/3 2/3 2/5 3/4 3/5 4/5 5/6 6/7 7/8 8/9 9/10 AUTO)
dvb.lsPol	CHOICE		EnumRange (HORIZONTAL VERTICAL)
dvb.lsModulation	CHOICE		EnumRange (BPSK OQPSK QPSK 8PSK 16APSK 16PSK 16QAM 32APSK 32QAM 64QAM 128QAM 256QAM AUTO)
dvb.lsDvbMode	CHOICE		EnumRange (DVB-S DVB-S2)
li	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dvb.ls.add	TEXT	NOPRESET	StringRange
dvb.ls.delete	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Summary warning)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary alarm)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.11	ALARM	R/O	AlarmFlagRange R/O (GPS)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Compass)
faults.13	ALARM	R/O	AlarmFlagRange R/O (SKYWAN)

9.412 NDSatCom-ACU5030

This page describes the device driver and the device window for the ND SatCom antenna control unit ACU5030.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Step track state, settings and beacon level
- --- Shows the satellite list (which can be configured via web interface of the controller). Move antenna to a preconfigured satellite.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	The ACU usually expects the address "A".

Network settings

Remember to configure the IP address, netmask and default gateway of the ACU5030 properly on the frontpanel. Then use the IP address and TCP port number 4010 to configure the device

interface. For example: 10.0.0.123:4010

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a	HEX	R/O	HexRange R/O (0 .. 0)
internal.b	HEX	R/O	HexRange R/O (0 .. 0)
internal.c	HEX	R/O	HexRange R/O (0 .. 0)
internal.d	HEX	R/O	HexRange R/O (0 .. 0)
internal.e	HEX	R/O	HexRange R/O (0 .. 0)
internal.u	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.v	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.w	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.x	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.y	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.z	BOOLEAN	R/O	BooleanRange R/O (true false)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0 .. 0)

name	*type*	*flags*	*range*
frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
trackMode	INTEGER	NOPRESET	IntegerRange (0 .. 0)
satellite	CHOICE	NOPRESET	EnumRange (UNKNOWN)
satname	TEXT	R/O	StringRange R/O
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
updateSatList	TEXT	NOPRESET	StringRange
edit.satellite	CHOICE		EnumRange (UNKNOWN)
edit.name	TEXT	R/O	StringRange R/O
edit.satNum	TEXT	R/O	StringRange R/O
edit.orbit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.polarization	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.bfreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.bpol	TEXT	R/O	StringRange R/O
edit.goto	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Beacon receive)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote control)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL interlock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (ODU communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (BCRX communication)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Peaking)

9.413 NDSatCom-ACU6030-REST

This page describes the REST-HTTP based device driver and the device window for the ND SatCom antenna control unit ACU6030. It provides exaclty the same scope of functions as the older (terminal protocol based) version of the driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Step track state, settings and beacon level
- --- Shows the satellite list (which can be configured via web interface of the controller). Move antenna to a preconfigured satellite.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	Enter the ACU's IP address here, including the pport nummber if this has to set to a non-standard value. Example: 10.10.0.1:9876

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.x	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.y	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.z	BOOLEAN	R/O	BooleanRange R/O (true false)
config.acuDeviceIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
azimuth	FLOAT		DoubleRange (0.0000 .. 0.0000)
elevation	FLOAT		DoubleRange (0.0000 .. 0.0000)
polarization	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
trackMode	INTEGER	NOPRESET	IntegerRange (0 .. 0)
satellite	CHOICE	NOPRESET	EnumRange (UNKNOWN)
satname	TEXT	R/O	StringRange R/O
state.az	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.el	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
updateSatList	TEXT	NOPRESET	StringRange
edit.satellite	CHOICE		EnumRange (UNKNOWN)
edit.name	TEXT	R/O	StringRange R/O
edit.satNum	TEXT	R/O	StringRange R/O
edit.orbit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.polarization	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.bfreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
edit.bpol	TEXT	R/O	StringRange R/O
edit.goto	TEXT	NOPRESET	StringRange
ai	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mGeneral	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Beacon receive)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote control)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL interlock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (ODU communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (BCRX communication)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Peaking)

9.414 NDSatCom-CPI

This page describes the device driver and the device window for the ND-SatCom TWT amplifiers running in CPI compatibility mode.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. It is recommended to operate the amplifier at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. When controlling an in-door unit or an out-door unit through the front panel, usually a RS232 interface is used (this is the factory setting of the front panel). The devices can however bridged for RS485 usage inside on the microprocessor control panel. The out-door unit always uses a RS485 interface.

- JP1 --- 1-2Tx Termination
- JP2 --- 1-2Rx Termination
- JP6* --- 1-22 wire RS485
- JP7* --- 1-24 wire RS485
- JP8* --- 1-2RS232

(*) Only one Jumper of the group should be closed

Installation

Before you can operate the amplifier with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.tx.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.checkLimits	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 40.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
meas.input	FLOAT	R/O	DoubleRange R/O (-100.0 .. 100.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 100.00)
meas.iBeam	INTEGER	R/O	IntegerRange R/O (0 .. 10000)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Standby timeout)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EEPROM)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.10	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External RF inhibit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RF-in overdrive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power supply thermal)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWT thermal)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low airflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.415 NDSatCom-KBand-Downconverter

This page describes the device driver and the device window for the ND-SatCom down-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Downconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Downconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Downconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Downconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the down-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hasRxOn	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange DIS (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (17300.000 .. 18400.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.416 NDSatCom-KBand-Upconverter

This page describes the device driver and the device window for the ND-SatCom up-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Upconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Upconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Upconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Upconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the up-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF"

special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (17300.000 .. 18400.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.417 NDSatCom-KuBand-Downconverter

This page describes the device driver and the device window for the ND-SatCom down-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Downconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Downconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Downconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Downconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the down-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hasRxOn	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange DIS (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.000 .. 12750.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.418 NDSatCom-KuBand-Upconverter

This page describes the device driver and the device window for the ND-SatCom up-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Upconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Upconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Upconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Upconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the up-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)

name	*type*	*flags*	*range*
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (12750.000 .. 14500.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.419 NDSatCom-KuHBand-Downconverter

This page describes the device driver and the device window for the ND-SatCom down-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Downconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Downconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Downconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Downconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the down-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.hasRxOn	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange DIS (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (11700.000 .. 12750.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.420 NDSatCom-KuHBand-Upconverter

This page describes the device driver and the device window for the ND-SatCom up-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Upconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Upconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Upconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Upconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the up-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (13750.000 .. 14500.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.421 NDSatCom-KuLBand-Downconverter

This page describes the device driver and the device window for the ND-SatCom down-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Downconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Downconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Downconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Downconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the down-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hasRxOn	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.000 .. 11700.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.422 NDSatCom-KuLBand-Upconverter

This page describes the device driver and the device window for the ND-SatCom up-converters. There are four different variants of the driver available which only differ at the RF frequency range:

- NDSatCom-KBand-Upconverter --- 17.3 .. 18.4 GHz
- NDSatCom-KuBand-Upconverter --- 12.75 .. 14.5 GHz
- NDSatCom-KuLBand-Upconverter --- 13.75 .. 14.5 GHz
- NDSatCom-KuHBand-Upconverter --- 12.75 .. 13.25 GHz

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. The factory preset setting of the up-converter is RS485, it however may be jumpered to RS232 inside.

Installation

Before you can operate the up-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.if	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (12750.000 .. 13250.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LO-A lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (LO-B lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IF-LO level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RF-LO level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.423 NDSatCom-RCU-RC11

This page describes the device driver and the device window for the ND-SatCom RCU redundancy controller via REST-API. It does not include the control of attached devices (like HPA or Modulator or Encoder) but is only responsible for the waveguide switch(es) and the redundancy settings and. Use the specific NDSatCom-TWTA-REST driver to access the attached devices via RCU ip address.

It supports only the following configuration:

- RC11 with a 1+1 redundancy (2 NDS HPA devices per chain, 1 Switch)

For support of other configurations see other drivers mentioned in the documentation.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Protections switching
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Controller.

parameter	***description***
address	The device IP address as set at the device's front panel followed by the API port which is set by default to 9876 from the vendor. E.g. 192.168.1.1:9876
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
operationMode	Select controllers switching mode SW-ONCE or SW-ALWAYS
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.deviceName.01	TEXT	R/O	StringRange R/O
info.deviceName.02	TEXT	R/O	StringRange R/O
info.deviceName.03	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.operationMode	CHOICE	SETUP	EnumRange (SW-ONCE SW-ALWAYS)
protection	CHOICE		EnumRange (MANUAL AUTO BLOCKED)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)
wgstat.1	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.1	HEX		HexRange (0 .. 0)
internal.model	TEXT	R/O	StringRange R/O
internal.dummy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.once	INTEGER	R/O	IntegerRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.loc	TEXT	R/O	StringRange R/O
mGeneral	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RCU Power Supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RCU Power Supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TX Chain 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (TX Chain 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (WG Switch 1)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unknown Version)

9.424 NDSatCom-RCU-RC11OMa

This page describes the device driver and the device window for the ND-SatCom RCU redundancy controller via REST-API. It does not include the control of attached devices (like HPA or Modulator or Encoder) but is only responsible for the waveguide switch(es) and the redundancy settings and. Use the specific NDSatCom-TWTA-REST driver to access the attached devices via RCU ip address.

It supports only the following configuration:

- RC1OMa with a 1+1 redundancy (2 devices per chain, 1 Switch) and Antenna/Load switch

For support of other configurations see other drivers mentioned in the documentation.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Protections switching
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the Controller.

parameter	***description***
address	The device IP address as set at the device's front panel followed by the API port which is set by default to 9876 from the vendor. E.g. 192.168.1.1:9876
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
operationMode	Select controllers switching mode SW-ONCE or SW-ALWAYS
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.deviceName.01	TEXT	R/O	StringRange R/O
info.deviceName.02	TEXT	R/O	StringRange R/O
info.deviceName.03	TEXT	R/O	StringRange R/O
info.deviceName.04	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.deviceName.05	TEXT	R/O	StringRange R/O
info.deviceName.06	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.operationMode	CHOICE	SETUP	EnumRange (SW-ONCE SW-ALWAYS)
protection	CHOICE		EnumRange (MANUAL AUTO BLOCKED)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)
wgstat.1	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.1	HEX		HexRange (0 .. 0)
location.2	CHOICE		EnumRange (LOAD ANTENNA)
position.2	CHOICE		EnumRange (A B)
wgstat.2	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.2	HEX		HexRange (0 .. 0)
internal.model	TEXT	R/O	StringRange R/O
internal.dummy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.once	INTEGER	R/O	IntegerRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
internal.loc	TEXT	R/O	StringRange R/O
internal2.loc	TEXT	R/O	StringRange R/O
mGeneral	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RCU Power Supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RCU Power Supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TX Chain 1)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (TX Chain 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (WG Switch 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (WG Switch 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unknown Version)

9.425 NDSatCom-RCU-SNMP

This page describes the device driver and the device window for the ND-SatCom RCU redundancy controller it does not include the control of attached devices (like HPA or ENC) but is only responsible for the waveguide switch(es) and the redundancy settings. Use the specific SNMP driver to access the attached devices via RCU ip address.

It supports at least the following configurations:

- BC02 with a 1+1 redundancy (3 devices per chain, 1 Switch) and Antenna/Load switch
- BC05 with a 1+1 redundancy (2 devices per chain, 1 Switch) and Antenna/Load switch
- BC11 with a 1+1 redundancy (2 devices per chain, 1 Switch)
- BC11s01 with a 1+1 redundancy (2 devices per chain, 1 Switch)
- BC13 with a 1+1 redundancy (1 device per chain, 1 Switch)
- BC20 with 1 1+1 redundancy (3 devices per chain, 1 Switch)
- BC42 with a 1+1 redundancy (2 devices per chain, 1 Switch)
- BC30 with a 1+1 redundancy (1 device per chain, 1 switch) and Antenna/Load switch and Polarization switch

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Protections switching
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*

parameter	*description*
address	The device P address as set at the device's front panel.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
operationMode	Select controllers switching mode SW-ONCE or SW-ALWAYS
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Communication protocol / Interface type

This device driver requires the SNMP protocol. The default community string is public for read-only and read-write.

Installation

Before you can operate the amplifier with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel configure IP address, netmask and gateway and use the default SNMP port number (161 UDP).

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.deviceName.01	TEXT	R/O	StringRange R/O DIS
info.deviceName.02	TEXT	R/O	StringRange R/O DIS
info.deviceName.03	TEXT	R/O	StringRange R/O DIS
info.deviceName.04	TEXT	R/O	StringRange R/O DIS
info.deviceName.05	TEXT	R/O	StringRange R/O DIS
info.deviceName.06	TEXT	R/O	StringRange R/O DIS
info.deviceState.01	TEXT	R/O	StringRange R/O DIS
info.deviceState.02	TEXT	R/O	StringRange R/O DIS
info.deviceState.03	TEXT	R/O	StringRange R/O DIS
info.deviceState.04	TEXT	R/O	StringRange R/O DIS
info.deviceState.05	TEXT	R/O	StringRange R/O DIS
info.deviceState.06	TEXT	R/O	StringRange R/O DIS
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.operationMode	CHOICE	SETUP	EnumRange (SW-ONCE SW-ALWAYS)
config.deviceComment	TEXT	SAVE SETUP	StringRange
protection	CHOICE		EnumRange DIS (Manual Auto Blocked)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange DIS (Chain1 Chain2)
position	CHOICE		EnumRange DIS (A B)
wgstat.1	HEX	R/O	HexRange R/O DIS (0 .. 0)
wgcmd.1	HEX		HexRange DIS (0 .. 0)
location.2	CHOICE		EnumRange DIS (Load Antenna)
position.2	CHOICE		EnumRange DIS (A B)
wgstat.2	HEX	R/O	HexRange R/O DIS (0 .. 0)
wgcmd.2	HEX		HexRange DIS (0 .. 0)

name	*type*	*flags*	*range*
location.3	CHOICE		EnumRange DIS (X Y)
position.3	CHOICE		EnumRange DIS (A B)
wgstat.3	HEX	R/O	HexRange R/O DIS (0 .. 0)
wgcmd.3	HEX		HexRange DIS (0 .. 0)
reset	TEXT	NOPRESET	StringRange
internal.sw11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.model	TEXT	R/O	StringRange R/O
internal.dummy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.once	INTEGER	R/O	IntegerRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
internal.loc	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RCU Warning)
faults.02	ALARM	R/O	AlarmFlagRange R/O (RCU)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TX Chain 1 Warning)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TX Chain 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (TX Chain 2 Warning)
faults.06	ALARM	R/O	AlarmFlagRange R/O (TX Chain 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Unknown model)
faults.11	ALARM	R/O	AlarmFlagRange R/O (WG Switch 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (WG Switch 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (WG Switch 3)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.426 NDSatCom-SkyWAN-IDU

This page describes the device driver and the device window for the NDSatCom SkyWAN IDU. The driver covers the models 7200 (software version 7.20) and 1070 (software versions 1.11 and 1.20). It monitors some general state variables and measurement values from the device, additionally it implements a command interface for the NDSatCom Media Fleet Manager interface available for the sat-nms software.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The contains the general transmit parameters of the IDU
- --- Displays meter readings.
- --- Contains same general parameters concerning QoS routing and also allows to debug the QoS settings commanded by a MFM interface extension, if installed and configured.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings must match the values set in the device. Common community strings are 'public' (read) and 'private' (write). Hence a typical address specification for a SkyWAN IDU may be *10.0.0.1 public private*.

- iduVersion --- Sets the IDU software version, this must match the software version running on the device.
- demods -- Sets the number of demodulator boards

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.chassisType.1	TEXT	R/O	StringRange R/O
info.chassisType.2	TEXT	R/O	StringRange R/O
info.chassisHwVer.1	TEXT	R/O	StringRange R/O
info.chassisHwVer.2	TEXT	R/O	StringRange R/O
info.chassisSwVer.1	TEXT	R/O	StringRange R/O
info.chassisSwVer.2	TEXT	R/O	StringRange R/O
info.chassisSerial.1	TEXT	R/O	StringRange R/O
info.chassisSerial.2	TEXT	R/O	StringRange R/O
info.softwareVersion	TEXT	R/O	StringRange R/O
info.softwareBuild	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.iduVersion	CHOICE	SAVE SETUP	EnumRange (1.20 7.20 1.11 7.11)

name	*type*	*flags*	*range*
config.demods	CHOICE	SAVE SETUP	EnumRange (1 2)
tdma.stationState	CHOICE		EnumRange (PASSIVE ACTIVE)
tdma.stationRtt	INTEGER		IntegerRange (0 .. 0)
tdma.stationRxFreqOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tdma.stationTxFreqOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tdma.txPathAttOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tdma.txPathAttBase	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tdma.homeChannel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tdma.stationAttnBase	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tdma.stationAttnOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tdma.stationAttn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tdma.localChannels	TEXT		StringRange
tdma.homeChannelTwo	INTEGER		IntegerRange (0 .. 0)
tdma.frameSyncState	CHOICE	R/O	EnumRange R/O (lost acquired)
tdma.slotSyncState	CHOICE	R/O	EnumRange R/O (lost acquired)
tdma.circularState	CHOICE	R/O	EnumRange R/O (awtngSllAddr awtngRangSlt usingRangSlt noSyncDataXc syncDataXchg slfReception)
tdma.circularGrade	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tdma.activeMaster	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tdma.backupMaster	INTEGER	R/O	IntegerRange R/O (0 .. 0)
demod.01.power	INTEGER	R/O	IntegerRange R/O (0 .. 0)
demod.02.power	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tpc.state	CHOICE	R/O	EnumRange R/O (tpcOffOrOk tooLow imprTooSmall txPathInComp tooHigh)
tpc.mode	CHOICE	R/O	EnumRange R/O (ENABLED DISABLED)

name	*type*	*flags*	*range*
tpc.esnoRef	FLOAT		DoubleRange (0.0 .. 0.0)
tpc.esnoRefValid	TEXT	R/O	StringRange R/O
tpc.esnoOwn	FLOAT		DoubleRange (0.0 .. 0.0)
tpc.esnoOwnValid	TEXT	R/O	StringRange R/O
tpc.esnoTarget	FLOAT		DoubleRange (0.0 .. 0.0)
tpc.esnoTargetValid	TEXT	R/O	StringRange R/O
qos.enabled	CHOICE		EnumRange (ENABLED DISABLED)
qos.dynamicMode	CHOICE		EnumRange (FLOW MICROFLOW MICROFLOW-EP)
addQosWeight1	TEXT		StringRange
addQosWeight2	TEXT		StringRange
addQosWeight3	TEXT		StringRange
addQosWeight4	TEXT		StringRange
addQosWeight5	TEXT		StringRange
addQosWeight6	TEXT		StringRange
addQosWeight7	TEXT		StringRange
addQosWeight8	TEXT		StringRange
delQosWeight1	TEXT		StringRange
delQosWeight2	TEXT		StringRange
delQosWeight3	TEXT		StringRange
delQosWeight4	TEXT		StringRange
delQosWeight5	TEXT		StringRange
delQosWeight6	TEXT		StringRange
delQosWeight7	TEXT		StringRange
delQosWeight8	TEXT		StringRange
qos.cmd	TEXT	R/O	StringRange R/O
qos.srcIP	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
qos.srcPort	TEXT	R/O	StringRange R/O
qos.dstIP	TEXT	R/O	StringRange R/O
qos.dstPort	TEXT	R/O	StringRange R/O
qos.dataRate	TEXT	R/O	StringRange R/O
qos.type	TEXT	R/O	StringRange R/O
qos.weight	TEXT	R/O	StringRange R/O
idv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state	CHOICE	R/O	EnumRange R/O (PASSIVE ACTIVE)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RX Lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transmit)

9.427 NDSatCom-Step-Track

This page describes the device driver and the device window for the NDSatCom step track system. The unit must be running a recent software version to be capable to be controlled by the VLC/M&C system.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.The set point parameters are shown with a resolution of 0.1 deg for azimuth/elevation and 1 deg for the polarization. This to make the spin buttons work in a convenient way. You still may enter azimuth/elevation values with a 0.01 deg resolution, the antenna controller accepts this. If you change the antenna pointing, tracking automatically stops.The "moving" LED doesn't work with this device, the antenna controller doesn't provide this information.
- --- The second parameter page displays/controls the tracking mode resp. state. Refer to the ND SatCom manual for details of the tracking modes in particular.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
address	The NDSatCom step track system does not use this parameter.
intTrckRecv	Set this to "INTERNAL" if the Step-Track-System contains the internal beacon receiver card.
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The NDSatCom step track expects *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.c	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.intTrckRecv	CHOICE	SAVE SETUP	EnumRange (NONE INTERNAL)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
polarization	FLOAT		DoubleRange (0 .. 0)
trackMode	CHOICE		EnumRange (STANDBY REMOTE POSITION MODEL TRACK (NORM) MODEL TRACK (LEARN) MODEL TRACK (PEAKING) SCAN BEACON)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
satNum	TEXT		StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.tracking	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
state.peaking	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU outdoor unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PL limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Volt meter)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote/peaking)

9.428 NDSatCom-TWTA-BC30

This page describes the device driver and the device window for the ND-SatCom TWT amplifiers controlled via SNMP and connected to the RCU version BC30 redundancy controller. There are

some limitations of this driver because of the provided MIB compared to drivers for the serial protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device IP address as set at the device's front panel.
hpaDeviceIndex	Normally 1 for a system with one TWTA, in other cases the number of the TWTA in the RCU configuration.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
checkLimits	En/Disables the monitoring of rf power limits
lowLimit	Sets the lower rf power alarm limit
highLimit	Sets the upper rf power alarm limit

Communication protocol / Interface type

This device driver requires the SNMP protocol. The default community string is public for read-only and read-write.

Installation

Before you can operate the amplifier with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel configure IP address, netmask and gateway and use the default SNMP port number (161 UDP).

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hpaDeviceIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.checkLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON COOLING+OFF STOP COOLING)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
state	TEXT	R/O	StringRange R/O
transmitSelected	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Standby timeout)
faults.06	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.09	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Low airflow)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply thermal)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.17	ALARM	R/O	AlarmFlagRange R/O (TWT thermal)
faults.18	ALARM	R/O	AlarmFlagRange R/O (BUC summary)
faults.19	ALARM	R/O	AlarmFlagRange R/O (BUC communication)
faults.20	ALARM	R/O	AlarmFlagRange R/O (HPA communication)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input Overdrive)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.429 NDSatCom-TWTA-REST

This page describes the device driver and the device window for the ND-SatCom TWT amplifiers controlled via REST-HTTP and connected to the RCU redundancy controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device IP address as set at the device's front panel.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
hpaDeviceIndex	Normally 1 for a system with one TWTA, in other cases the number of the TWTA in the RCU configuration.
bucReference	Reference source for the integrated BUC if available.
checkLimits	En/Disables the monitoring of rf power limits
lowLimit	Sets the lower rf power alarm limit
highLimit	Sets the upper rf power alarm limit
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes are logged in the event database even when LogParameterChanges and LogDetectedChanges are switched ON. It is intended to be en/disabled by an external UPC.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.bucInstalled	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.triBandSupport	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwRevision	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.fwRelease	TEXT	R/O	StringRange R/O
info.heaterHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamOnHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.maxPower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hpaDeviceIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bucReference	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.checkLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.coolingTime	INTEGER	SETUP	IntegerRange (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
state	TEXT	R/O	StringRange R/O
tx.alc	CHOICE		EnumRange (OFF ON)
tx.requestedLevel	FLOAT		DoubleRange (-30.0 .. 30.0)
tx.coolingOff	TEXT	NOPRESET	StringRange
tx.stopCoolingOff	TEXT	NOPRESET	StringRange
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
hi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.maxOutputPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
int.state	TEXT	R/O	StringRange R/O
int.tSelected	TEXT	R/O	StringRange R/O
faults.02	ALARM	R/O	AlarmFlagRange R/O (BUC communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (HPA communication)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.10	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RF-in overdrive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power supply thermal)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWT thermal)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low airflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Timeout standby)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.430 NDSatCom-TWTA-SNMP

This page describes the device driver and the device window for the ND-SatCom TWT amplifiers controlled via SNMP and connected to the RCU redundancy controller. There are some limitations of this driver because of the provided MIB compared to drivers for the serial protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device IP address as set at the device's front panel.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
hpaDeviceIndex	Normally 1 for a system with one TWTA, in other cases the number of the TWTA in the RCU configuration.
bucReference	Reference source for the integrated BUC if available.
checkLimits	En/Disables the monitoring of rf power limits
lowLimit	Sets the lower rf power alarm limit
highLimit	Sets the upper rf power alarm limit
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes are logged in the event database even when LogParameterChanges and LogDetectedChanges are switched ON. It is intended to be en/disabled by an external UPC.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Communication protocol / Interface type

This device driver requires the SNMP protocol. The default community string is public for read-only and read-write.

Installation

Before you can operate the amplifier with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel configure IP address, netmask and gateway and use the default SNMP port number (161 UDP).

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.bucInstalled	CHOICE	R/O	EnumRange R/O (NO YES)
info.model	CHOICE		EnumRange (Ku-Band K-Band Ka-Band C-Band X-Band)
info.triBandSupport	CHOICE	R/O	EnumRange R/O (NO YES)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.hpaDeviceIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bucReference	CHOICE	SETUP	EnumRange DIS (internal external auto)
config.checkLimits	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON COOLING+OFF STOP COOLING)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
state	TEXT	R/O	StringRange R/O
transmitSelected	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.coolingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
faults.02	ALARM	R/O	AlarmFlagRange R/O (BUC communication)
faults.03	ALARM	R/O	AlarmFlagRange R/O (HPA communication)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF inhibit)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.10	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Helix over current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power supply thermal)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWT thermal)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low airflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Timeout standby)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.431 NDSatCom-TWTA

This page describes the device driver and the device window for the ND-SatCom 350/700W TWT amplifiers. Actually the *NDSatCom-TWTA* driver is available which covers the 300/700 Watts 14 and 18 GHz variants of the amplifier.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
checkLimits	Turns the amplifier's built in power limit check off or on. When turned on, the amplifier generates a fault if the output power is outside the limits set below.
lowLimit	The low output power limit.
highLimit	The high output power limit.
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
logGainChanges	Defines if gain changes are logged in the event database even when LogParameterChanges and LogDetectedChanges are switched ON. It is intended to be en/disabled by an external UPC.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. It is recommended to operate the amplifier at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Miteq-MOD95 protocol. When controlling an in-door unit or an out-door unit through the front panel, usually a RS232 interface is used (this is the factory setting of the front panel). The devices can however jumpered for RS485 usage inside on the microprocessor control panel. The out-door unit always uses a RS485 interface.

- JP1 --- 1-2Tx Termination

- JP2 --- 1-2Rx Termination
- JP6* --- 1-22 wire RS485
- JP7* --- 1-24 wire RS485
- JP8* --- 1-2RS232

(*) Only one Jumper of the group should be closed

Installation

Before you can operate the amplifier with the sat-nms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, use the "SF" special functions to set the unit to 9600 baud / no parity. The device address must match the value set at the configuration parameters above.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.tx.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.checkLimits	CHOICE	SETUP	EnumRange (OFF ON)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)

name	*type*	*flags*	*range*
config.linearizer	FLOAT	SETUP	DoubleRange (0.0 .. 20.0)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 40.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (-100.0 .. 40.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
meas.input	FLOAT	R/O	DoubleRange R/O (-100.0 .. 100.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 100.00)
meas.iHeater	INTEGER	R/O	IntegerRange R/O (0 .. 10000)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 999.9)
internal.dummy	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.summ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.bsl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lcl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.sec1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Standby timeout)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.07	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.10	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal interlock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External RF inhibit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Helix avg. over current)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Heater over current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply arc trip)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (RF-in overdrive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power supply thermal)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWT thermal)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low airflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.432 NMEA-GPS-Receiver

This page describes the device driver and the device window for GPS receivers offering the NMEA 0183 protocol through a serial interface.

The sat-nms device driver for this device family passively reads the data stream output by the GPS receiver. From the received data the driver extracts the GGA (Global Positioning System Fix Data) sentence and parses it.

Important Notice

Due to the fact that the GPS receiver perpetually sends it's data to the VLC, it is essential that the *idle time* of the device thread / interface serving the GPS receiver is set to 10 msecs. The default value of this parameter is 1000 msecs which leads to frequent read timeouts / misinterpretations due to overruns of the VLC serial interface input buffer. The idle time parameter may be changed with the interface properties at the device setup editor.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page shows the GPS parameters read from the device.
- --- The faults page.
- --- The device info page.

- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The only configuration parameter which must be set is the line speed of the serial interface. GPS receivers usually work at *4800,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.gga	TEXT	R/O	StringRange R/O
longitude	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
latitude	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
utcTime	TEXT	R/O	StringRange R/O
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (GPS receive)

9.433 NPC-ACU-550

This page describes the device driver and the device window for the NPC SYSTEMS NEYRPIC ACU 550 via SNMP. The driver was tested with the following versions:

- APP Version: 1.20.02
- IHM Version: 1.45
- SNMP Version: 1.10

As the SNMP interface does not allow to read the list of stored targets on the ACU it is only possible to load a preset position or tracking using the storage numbers 0-63.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the tracking parameters and state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The SNMP address specification for the device (details below).
hasPolAxis	Defines id the antenna is equipped with a pol axis drive.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	***description***
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.

parameter	*description*
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.appVersion	TEXT	R/O	StringRange R/O
info.webVersion	TEXT	R/O	StringRange R/O
info.ihmVersion	TEXT	R/O	StringRange R/O
info.paramVersion	TEXT	R/O	StringRange R/O
info.snmpVersion	TEXT	R/O	StringRange R/O
info.longitude	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.latitude	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.altitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.stationName	TEXT	R/O	StringRange R/O
info.receiver	TEXT	R/O	StringRange R/O
info.polarization	TEXT	R/O	StringRange R/O
info.controlMode	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.switches	INTEGER	R/O	IntegerRange R/O (0 .. 6)
internal.plSwitches	INTEGER	R/O	IntegerRange R/O (0 .. 2)
internal.azErr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.elErr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.no	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.hasPolAxis	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.deviceComment	TEXT	SAVE SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azPosErr	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elPosErr	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.plPosErr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.azSpeed	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elSpeed	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.plSpeed	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
step	TEXT	NOPRESET	StringRange
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.acqLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.target	TEXT	R/O	StringRange R/O
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
target.no	INTEGER		IntegerRange (0 .. 63)
prespos.no	INTEGER		IntegerRange (0 .. 63)
trackMode	CHOICE		EnumRange (STANDBY MANUAL STEP TRACK SMART ORBITAL MONOPULSE ORBITAL AUTO TRACK)
searchType	TEXT	NOPRESET	StringRange
state.searchType	CHOICE	R/O	EnumRange R/O (OFF ON ON-ACQ)
phasing	CHOICE		EnumRange (OFF ON FORCE)
signalTrack	CHOICE		EnumRange (OFF ON)
mp.xelevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elevationError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azErr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.elErr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reboot	TEXT	NOPRESET	StringRange
ackFaults	TEXT	NOPRESET	StringRange
stow	TEXT	NOPRESET	StringRange
unstow	TEXT	NOPRESET	StringRange
switch.01	CHOICE		EnumRange (A B)
switch.02	CHOICE		EnumRange (A B)
switch.03	CHOICE		EnumRange (A B)
switch.04	CHOICE		EnumRange (A B)
switch.05	CHOICE		EnumRange (A B)
switch.06	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary Warning)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Maintenance Mode)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Main PS)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Aux PS)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ motor)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EL motor)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PL motor)
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ cw hard limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ ccw hard limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL up hard limit)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (EL dn hard limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PL cw hard limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PL ccw hard limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (AZ cw soft limit)
faults.16	ALARM	R/O	AlarmFlagRange R/O (AZ ccw soft limit)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL up soft limit)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL dn soft limit)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PL cw soft limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PL ccw soft limit)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Receiver)
faults.22	ALARM	R/O	AlarmFlagRange R/O (RX lock)
faults.23	ALARM	R/O	AlarmFlagRange R/O (RX path)
faults.24	ALARM	R/O	AlarmFlagRange R/O (AZ Zone)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Autocal)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Lubrication)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Signal lost)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Signal too strong)
faults.29	ALARM	R/O	AlarmFlagRange R/O (AZ immobile)
faults.30	ALARM	R/O	AlarmFlagRange R/O (EL immobile)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PL immobile)
faults.32	ALARM	R/O	AlarmFlagRange R/O (AZ reverse)
faults.33	ALARM	R/O	AlarmFlagRange R/O (EL reverse)

name	*type*	*flags*	*range*
faults.34	ALARM	R/O	AlarmFlagRange R/O (PL reverse)
faults.35	ALARM	R/O	AlarmFlagRange R/O (AZ runaway)
faults.36	ALARM	R/O	AlarmFlagRange R/O (EL runaway)
faults.37	ALARM	R/O	AlarmFlagRange R/O (PL runaway)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Autophase table)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Monopulse)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Stow lock)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Optimization)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Time setting)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Synchro)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Mode suspended)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.46	ALARM	R/O	AlarmFlagRange R/O (RX in local)
faults.47	ALARM	R/O	AlarmFlagRange R/O (No orital param)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Stow mode req)
faults.49	ALARM	R/O	AlarmFlagRange R/O (End estim valid)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Ephermid obsolete)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Start too early)
faults.52	ALARM	R/O	AlarmFlagRange R/O (AZ range too large)
faults.53	ALARM	R/O	AlarmFlagRange R/O (No ephemerid)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Trajectory)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Trajectory incorr)

name	*type*	*flags*	*range*
faults.56	ALARM	R/O	AlarmFlagRange R/O (RF inibit)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Switch 6)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Local Control)

9.434 NTT-HVD6100

This page describes the device driver and the device window for the NTT HVD6100 IRD with support for RF input (DVB-S/S2), ASI input and IP input.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The main receive parameters, including frequency and service selection.
- --- IP parameters
- --- The conditional access parameters
- --- The video parameters
- --- The audio parameters
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the front panel or the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for a NTT-HVD6100 may be *10.0.0.1 public private*.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized to a proved preset value which should not be changed under normal circumstances.
- 22kHz --- Turns the 22 kHz signal in the LNB supply on or off. Only effective if 22kHzAuto is set OFF.
- 22kHzAuto --- Switching this on, makes the driver controlling the 22kHz signal with the receive frequency. For frequencies above the "band edge" 22kHz gets switched on.
- lnbPower --- Switches the LNB power supply ON or OFF.
- lnbVoltage --- Selects the LNB voltage (13V/18V) or switches it OFF. Only effective if the "lnbPolSwitch" parameter is set OFF.
- lnbPolSwitch --- Selects if the LNB voltage shall be permanently set to the the value selected by "lnbVoltage" (OFF) or if the voltage shall be selected from the "polarization" parameter of the driver.
- loFreqHigh --- The high band LO frequency.
- loFreqLow --- The low band LO frequency.
- bandEdge --- The band edge frequency.
- marginThreshold --- If "state.margin" drops below this value, a "Margin" fault is generated.
- serviceHunt --- Setting this ON lets the receiver use the first available service when receiving a new signal. OFF forces the receiver to get the service with the service ID set.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.temperature1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.mod	TEXT	R/O	StringRange R/O
internal.dvbmode	TEXT	R/O	StringRange R/O
internal.s2mode	TEXT	R/O	StringRange R/O
internal.va	TEXT	R/O	StringRange R/O
internal.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.input	TEXT	R/O	StringRange R/O
internal.tuner	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
input	CHOICE		EnumRange (ASI RF1 RF2 IP)
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
s2mode	CHOICE		EnumRange (AUTO MANUAL VCM)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE		EnumRange (none QPSK 8PSK 16APSK)
fec	CHOICE		EnumRange (none 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 8/9 9/10)
pls	INTEGER		IntegerRange (0 .. 262141)
searchRange	INTEGER		IntegerRange (0 .. 0)
polarization	CHOICE		EnumRange (Y X)
rolloff	CHOICE		EnumRange (35% 25% 20%)
pilot	CHOICE		EnumRange (none OFF ON)
packetLength	CHOICE		EnumRange (none SHORT NORMAL)
programMode	CHOICE		EnumRange (AUTO MANUAL)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.genlockMode	CHOICE		EnumRange (BYPASS FS-ONLY FS-SRC)

name	*type*	*flags*	*range*
video.genlockHorzAdj	INTEGER		IntegerRange (-32768 .. 32767)
video.genlockVerzAdj	INTEGER		IntegerRange (-32768 .. 32767)
video.genlockDelay	CHOICE		EnumRange (0 FRAME 0.5 FRAME 0.7 FRAME 1 FRAME)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.standard	CHOICE	R/O	EnumRange R/O (H.264 MPEG-2 UNKNOWN NONE)
video.format	TEXT	R/O	StringRange R/O
video.profile	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
video.fail	CHOICE		EnumRange (DISABLE ENABLE)
video.freezeDelay	CHOICE		EnumRange (0sec 30sec 60sec DISABLE SDI OFFSIGNAL)
video.rgbOutput	CHOICE		EnumRange (YPbPr RGB)
video.cvbsPedestal	CHOICE		EnumRange (0% IRE 7.5% IRE)
video.convert	CHOICE		EnumRange (to 480i29 to 576i25 to 1080i25)
video.arDownConvert	CHOICE		EnumRange (PAN SCAN LETTERBOX FULL FRAME)
video.horizontalPosition	INTEGER		IntegerRange (-100 .. 100)
output.conn1	CHOICE		EnumRange (ORIGINAL SDI CONVERSION SDI ASI)
output.conn2	CHOICE		EnumRange (ORIGINAL SDI CONVERSION SDI ASI)
output.conn3	CHOICE		EnumRange (ORIGINAL SDI CONVERSION SDI ASI)
output.tsOutput	CHOICE		EnumRange (ASI TUNER TS over IP POST TS DESCRAMBLE)
audio.autoPid	CHOICE		EnumRange (AUTO MANUAL)
audio.1.delay	CHOICE		EnumRange (0 FRAME 1 FRAME)

name	*type*	*flags*	*range*
audio.1.pid	INTEGER		IntegerRange (0 .. 0)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.downmix	CHOICE		EnumRange (NO YES)
audio.1.sampling	CHOICE	R/O	EnumRange R/O (16 bit 20 bit NONE)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.dolbyDecode	CHOICE		EnumRange (DECODE PASS-THROUGH)
audio.1.output	TEXT	R/O	StringRange R/O
audio.2.delay	CHOICE		EnumRange (0 FRAME 1 FRAME)
audio.2.pid	INTEGER		IntegerRange (0 .. 0)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.downmix	CHOICE		EnumRange (NO YES)
audio.2.sampling	CHOICE	R/O	EnumRange R/O (16 bit 20 bit NONE)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.dolbyDecode	CHOICE		EnumRange (DECODE PASS-THROUGH)
audio.2.output	TEXT	R/O	StringRange R/O
audio.3.delay	CHOICE		EnumRange (0 FRAME 1 FRAME)
audio.3.pid	INTEGER		IntegerRange (0 .. 0)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.downmix	CHOICE		EnumRange (NO YES)
audio.3.sampling	CHOICE	R/O	EnumRange R/O (16 bit 20 bit NONE)
audio.3.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.3.dolbyDecode	CHOICE		EnumRange (DECODE PASS-THROUGH)
audio.3.output	TEXT	R/O	StringRange R/O
audio.4.delay	CHOICE		EnumRange (0 FRAME 1 FRAME)
audio.4.pid	INTEGER		IntegerRange (0 .. 0)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.downmix	CHOICE		EnumRange (NO YES)
audio.4.sampling	CHOICE	R/O	EnumRange R/O (16 bit 20 bit NONE)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.dolbyDecode	CHOICE		EnumRange (DECODE PASS-THROUGH)
audio.4.output	TEXT	R/O	StringRange R/O
ipin.addressType	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (0 .. 0)
ipin.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.enable	CHOICE		EnumRange (DISABLE ENABLE)
ipout.addressType	CHOICE		EnumRange (MULTICAST UNICAST)
ipout.address	TEXT		StringRange
ipout.port	INTEGER		IntegerRange (0 .. 0)
ipout.protocol	CHOICE		EnumRange (UDP RTP)
ipout.packets	CHOICE		EnumRange (1 2 3 4 5 6 7)
ipout.ttl	INTEGER		IntegerRange (1 .. 255)

name	*type*	*flags*	*range*
ipout.serviceType	CHOICE		EnumRange (NORMAL DELAY MAX THROUGHPUT MAX RELIABILITY MIN MONETARY COST)
ipout.source	CHOICE		EnumRange (ASI TUNER TS over IP POST TS DESCRAMBLE)
ipout.fecEnable	CHOICE		EnumRange (DISABLE ENABLE)
ipout.fecMode	CHOICE		EnumRange (1D 5x5 1D 5x20 1D 10x10 2D 5x5 2D 5x20 2D 10x10)
ipout.fecAlignment	CHOICE		EnumRange (ANNEX A ANNEX B)
ipout.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ip	BOOLEAN	R/O	BooleanRange R/O (true false)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.spectral	TEXT	R/O	StringRange R/O
state.ber	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
ca.bisskey	TEXT	NOCOMPARE	StringRange
ca.injectedid	TEXT	NOCOMPARE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (TS over IP)
faults.05	ALARM	R/O	AlarmFlagRange R/O (ASI)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Warning)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temperature 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (FAN 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (FAN 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (FAN 3)
faults.18	ALARM	R/O	AlarmFlagRange R/O (FAN 4)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.435 NTT-HVE9100

This page describes the device driver and the device window for the NTT HVE9100 video encoder family without internal DVB-S/S2 modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- MPEG output parameters
- --- C/A encryption parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- IP parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and 'private' as write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for this device may be *10.0.0.1 public private*.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.temperature1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.tsclock	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL WAITING)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.setDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 10.0)
internal.vc	CHOICE		EnumRange (MPEG2 MPEG4-H264)
internal.ip	TEXT	R/O	StringRange R/O
internal.a1	TEXT	R/O	StringRange R/O
internal.a2	TEXT	R/O	StringRange R/O
internal.a3	TEXT	R/O	StringRange R/O
internal.a4	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
output.clock	CHOICE		EnumRange (INTERNAL EXTERNAL)
output.format	CHOICE		EnumRange (PACKET BURST)
output.packetsize	CHOICE		EnumRange (188 204)
output.datarate	FLOAT		DoubleRange (0.000 .. 0.000)
output.latency	CHOICE		EnumRange (STANDARD LOW RESERVED MEDIUM SUPERLOW)
output.netname	TEXT		StringRange

name	*type*	*flags*	*range*
output.netid	INTEGER		IntegerRange (0 .. 65535)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.tsid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (16 .. 8190)
output.pcrpid	INTEGER		IntegerRange (16 .. 8190)
output.nitpid	INTEGER		IntegerRange (16 .. 8190)
output.sdtid	INTEGER		IntegerRange (16 .. 8190)
output.nit.Enabled	CHOICE		EnumRange (DISABLE ENABLE)
output.sdt.Enabled	CHOICE		EnumRange (DISABLE ENABLE)
output.provider	TEXT		StringRange
output.service	TEXT		StringRange
output.encodeDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E)
ca.bisskey	TEXT	NOPRESET	StringRange
ca.injectedid	TEXT	NOPRESET	StringRange
video.coding	CHOICE		EnumRange (MPEG2 MPEG4-H264)
video.profile	CHOICE		EnumRange (MAIN 422P HIGH HIGH422)
video.resolution	CHOICE		EnumRange (1080i25 1080i29 1080i30 720p50 720p59 720p60 480i29 576i25)
video.entropy	CHOICE		EnumRange (N/A CAVLC CABAC)
video.filter	CHOICE		EnumRange (AUTO MANUAL)
video.filterStrength	CHOICE		EnumRange (OFF LIGHT MEDIUM HEAVY)
video.filterTemporal	CHOICE		EnumRange (OFF ON)
video.aspect	CHOICE		EnumRange (4:3 16:9)
video.size	CHOICE		EnumRange (1920 1440)

name	*type*	*flags*	*range*
video.bitrateMode	CHOICE		EnumRange (AUTO MANUAL)
video.source	CHOICE		EnumRange (SDI1 SDI2 AUTO)
video.datarate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.paff	CHOICE		EnumRange (OFF ON)
video.loss	CHOICE		EnumRange (LASTFRAME NULLOUTPUT)
video.test	CHOICE		EnumRange (OFF BARS)
audio.test	CHOICE		EnumRange (OFF SIN18dBFS SIN20dBFS)
video.streamid	INTEGER		IntegerRange (224 .. 239)
video.pid	INTEGER		IntegerRange (16 .. 8190)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.gopStruct	CHOICE		EnumRange (IP IBP IBBP)
video.gopLength	INTEGER		IntegerRange (1 .. 30)
video.gopMode	CHOICE		EnumRange (AUTO MANUAL)
video.gopType	CHOICE		EnumRange (N/A IDR FIRST GOP IDR EVERY GOP)
video.closedGop	CHOICE		EnumRange (N/A ON OFF)
audio.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.coding	CHOICE		EnumRange (NONE MPEG1 STEREO SMPTE302M 2CH DOLBY-D-PASS MPEG1 MONO MPEG1 DUAL MONO)
audio.1.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.1.delay	INTEGER		IntegerRange (-100 .. 50)
audio.1.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.1.pid	INTEGER		IntegerRange (16 .. 8190)
audio.1.streamId	INTEGER		IntegerRange (192 .. 223)
audio.1.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)

name	*type*	*flags*	*range*
audio.2.coding	CHOICE		EnumRange (NONE MPEG1 STEREO SMPTE302M 2CH DOLBY-D-PASS MPEG1 MONO MPEG1 DUAL MONO)
audio.2.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.2.delay	INTEGER		IntegerRange (-100 .. 50)
audio.2.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.2.pid	INTEGER		IntegerRange (16 .. 8190)
audio.2.streamId	INTEGER		IntegerRange (192 .. 223)
audio.2.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)
audio.3.coding	CHOICE		EnumRange (NONE MPEG1 STEREO SMPTE302M 2CH DOLBY-D-PASS MPEG1 MONO MPEG1 DUAL MONO)
audio.3.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.3.delay	INTEGER		IntegerRange (-100 .. 50)
audio.3.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.3.pid	INTEGER		IntegerRange (16 .. 8190)
audio.3.streamId	INTEGER		IntegerRange (192 .. 223)
audio.3.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)
audio.4.coding	CHOICE		EnumRange (NONE MPEG1 STEREO SMPTE302M 2CH DOLBY-D-PASS MPEG1 MONO MPEG1 DUAL MONO)
audio.4.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.4.delay	INTEGER		IntegerRange (-100 .. 50)
audio.4.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.4.pid	INTEGER		IntegerRange (16 .. 8190)
audio.4.streamId	INTEGER		IntegerRange (192 .. 223)
audio.4.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video 1 input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Encode status)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Warning)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.16	ALARM	R/O	AlarmFlagRange R/O (FAN)

9.436 NTT-HVE9210

This page describes the device driver and the device window for the NTT HVE9210 video encoder family without internal DVB-S/S2 modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- MPEG output parameters
- --- C/A encryption parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- Audio 3 parameters
- --- Audio 4 parameters
- --- IP parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and 'private' as write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for a NTT-HVE9210 may be *10.0.0.1 public private* .

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.temperature1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.tsclock	CHOICE	R/O	EnumRange R/O (INTERNAL EXTERNAL WAITING)
info.ipchannel	TEXT	R/O	StringRange R/O
info.ipchannel.1	TEXT	R/O	StringRange R/O
info.ipchannel.2	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.setDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 10.0)
internal.va	TEXT	R/O	StringRange R/O
internal.ip	TEXT	R/O	StringRange R/O
internal.a1	TEXT	R/O	StringRange R/O
internal.a2	TEXT	R/O	StringRange R/O
internal.a3	TEXT	R/O	StringRange R/O
internal.a4	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
output.clock	CHOICE		EnumRange (INTERNAL EXTERNAL)
output.format	CHOICE		EnumRange (PACKET BURST)
output.packetsize	CHOICE		EnumRange (188 204)
output.datarate	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
output.latency	CHOICE		EnumRange (STANDARD LOW MEDIUM SUPERLOW ULTRALOW)
output.netname	TEXT		StringRange
output.netid	INTEGER		IntegerRange (0 .. 65535)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.tsid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (16 .. 8190)
output.pcrpid	INTEGER		IntegerRange (16 .. 8190)
output.nitpid	INTEGER		IntegerRange (16 .. 8190)
output.sdtid	INTEGER		IntegerRange (16 .. 8190)
output.nit.Enabled	CHOICE		EnumRange (DISABLE ENABLE)
output.sdt.Enabled	CHOICE		EnumRange (DISABLE ENABLE)
output.provider	TEXT		StringRange
output.service	TEXT		StringRange
output.encodeDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.profile	CHOICE		EnumRange (MAIN HIGH HIGH422)
video.resolution	CHOICE		EnumRange (1080i25 1080i29 1080i30 1080p23 1080p24 1080p50 1080p59 1080p60 720p50 720p59 720p60 576i25 480i29 1080p29 1080p30)
video.lowRateMode	CHOICE		EnumRange (DISABLE ENABLE)
video.entropy	CHOICE		EnumRange (CAVLC CABAC)
video.filter	CHOICE		EnumRange (AUTO MANUAL)
video.filterStrength	CHOICE		EnumRange (OFF ADAPTIVE LIGHT MEDIUM HEAVY)
video.aspect	CHOICE		EnumRange (4:3 16:9(SQUEEZE) 16:9(LETTER BOX))
video.size	CHOICE		EnumRange (1920 1440 1280 960)

name	*type*	*flags*	*range*
video.sizeSD	CHOICE		EnumRange (720 352)
video.bitrateMode	CHOICE		EnumRange (AUTO MANUAL)
video.rateControl	CHOICE		EnumRange (CBR UNSTUFFING-CBR)
video.seamlessMode	CHOICE		EnumRange (DISABLE ENABLE)
video.source	CHOICE		EnumRange (SDI1 SDI2 AUTO)
video.datarate	INTEGER		IntegerRange (0 .. 0)
video.loss	CHOICE		EnumRange (LASTFRAME NULLOUTPUT)
video.test	CHOICE		EnumRange (DISABLE ENABLE)
video.testTon	CHOICE		EnumRange (DISABLE SIN18dBFS SIN20dBFS)
video.streamid	INTEGER		IntegerRange (224 .. 239)
video.pid	INTEGER		IntegerRange (16 .. 8190)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.gopStruct	CHOICE		EnumRange (IP IBP IBBP)
video.gopLength	INTEGER		IntegerRange (1 .. 30)
video.gopMode	CHOICE		EnumRange (AUTO MANUAL)
video.gopType	CHOICE		EnumRange (DISABLE ENABLE)
audio.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
audio.1.coding	CHOICE		EnumRange (NONE MPEG1 MONO MPEG1 STEREO MPEG1 DUAL MONO MPEG2 AAC-LC 1/0 MPEG2 AAC-LC 2/0 MPEG2 AAC-LC 1+1 MPEG2 AAC-LC 3/2/1 MPEG2 AAC-LC 4/0 MPEG2 AAC-LC 6/0 SMPTE302M 2CH SMPTE302M 4CH SMPTE302M 6CH SMPTE302M 8CH MPEG4 AAC-LC 2/0 MPEG4 AAC-LC 3/2/1 MPEG4 HE-AAC 2/0 MPEG4 HE-AAC 3/2/1 DOLBY-D-PASS DOLBY-DIGITAL-1/0 DOLBY-DIGITAL-2/0 AAC-LD-2/0 AAC-LD-3/1 AAC-LD-3/2/1 AAC-ELD-2/0 AAC-ELD-3/1 AAC-ELD-3/2/1)
audio.1.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.1.delay	INTEGER		IntegerRange (-100 .. 50)
audio.1.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.1.pid	INTEGER		IntegerRange (16 .. 8190)
audio.1.streamId	INTEGER		IntegerRange (192 .. 223)
audio.1.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)
audio.2.coding	CHOICE		EnumRange (NONE MPEG1 MONO MPEG1 STEREO MPEG1 DUAL MONO MPEG2 AAC-LC 1/0 MPEG2 AAC-LC 2/0 MPEG2 AAC-LC 1+1 MPEG2 AAC-LC 3/2/1 MPEG2 AAC-LC 4/0 MPEG2 AAC-LC 6/0 SMPTE302M 2CH SMPTE302M 4CH SMPTE302M 6CH SMPTE302M 8CH MPEG4 AAC-LC 2/0 MPEG4 AAC-LC 3/2/1 MPEG4 HE-AAC 2/0 MPEG4 HE-AAC 3/2/1 DOLBY-D-PASS DOLBY-DIGITAL-1/0 DOLBY-DIGITAL-2/0 AAC-LD-2/0 AAC-LD-3/1 AAC-LD-3/2/1 AAC-ELD-2/0 AAC-ELD-3/1 AAC-ELD-3/2/1)
audio.2.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.2.delay	INTEGER		IntegerRange (-100 .. 50)

name	*type*	*flags*	*range*
audio.2.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.2.pid	INTEGER		IntegerRange (16 .. 8190)
audio.2.streamId	INTEGER		IntegerRange (192 .. 223)
audio.2.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)
audio.3.coding	CHOICE		EnumRange (NONE MPEG1 MONO MPEG1 STEREO MPEG1 DUAL MONO MPEG2 AAC-LC 1/0 MPEG2 AAC-LC 2/0 MPEG2 AAC-LC 1+1 MPEG2 AAC-LC 3/2/1 MPEG2 AAC-LC 4/0 MPEG2 AAC-LC 6/0 SMPTE302M 2CH SMPTE302M 4CH SMPTE302M 6CH SMPTE302M 8CH MPEG4 AAC-LC 2/0 MPEG4 AAC-LC 3/2/1 MPEG4 HE-AAC 2/0 MPEG4 HE-AAC 3/2/1 DOLBY-D-PASS DOLBY-DIGITAL-1/0 DOLBY-DIGITAL-2/0 AAC-LD-2/0 AAC-LD-3/1 AAC-LD-3/2/1 AAC-ELD-2/0 AAC-ELD-3/1 AAC-ELD-3/2/1)
audio.3.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.3.delay	INTEGER		IntegerRange (-100 .. 50)
audio.3.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.3.pid	INTEGER		IntegerRange (16 .. 8190)
audio.3.streamId	INTEGER		IntegerRange (192 .. 223)
audio.3.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)

name	*type*	*flags*	*range*
audio.4.coding	CHOICE		EnumRange (NONE MPEG1 MONO MPEG1 STEREO MPEG1 DUAL MONO MPEG2 AAC-LC 1/0 MPEG2 AAC-LC 2/0 MPEG2 AAC-LC 1+1 MPEG2 AAC-LC 3/2/1 MPEG2 AAC-LC 4/0 MPEG2 AAC-LC 6/0 SMPTE302M 2CH SMPTE302M 4CH SMPTE302M 6CH SMPTE302M 8CH MPEG4 AAC-LC 2/0 MPEG4 AAC-LC 3/2/1 MPEG4 HE-AAC 2/0 MPEG4 HE-AAC 3/2/1 DOLBY-D-PASS DOLBY-DIGITAL-1/0 DOLBY-DIGITAL-2/0 AAC-LD-2/0 AAC-LD-3/1 AAC-LD-3/2/1 AAC-ELD-2/0 AAC-ELD-3/1 AAC-ELD-3/2/1)
audio.4.bitrate	CHOICE		EnumRange (128 kbps 192 kbps)
audio.4.delay	INTEGER		IntegerRange (-100 .. 50)
audio.4.extDelay	CHOICE		EnumRange (0 FRAME 1 FRAME 2 FRAME)
audio.4.pid	INTEGER		IntegerRange (16 .. 8190)
audio.4.streamId	INTEGER		IntegerRange (192 .. 223)
audio.4.samplingRate	CHOICE		EnumRange (16 BIT 20 BIT)
ipout.mode	CHOICE		EnumRange (DISABLE CH1 CH2 CH1-CH2)
ipout.ttl	INTEGER		IntegerRange (1 .. 125)
ipout.tos	INTEGER		IntegerRange (0 .. 255)
ipout.1.addressType	CHOICE		EnumRange (UNICAST MULTICAST)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (1024 .. 65530)
ipout.1.protocol	CHOICE		EnumRange (RTP UDP)
ipout.1.bitrate	INTEGER		IntegerRange (0 .. 0)
ipout.1.fecMode	CHOICE		EnumRange (DISABLE PRO-MPEG-1D-20X5 PRO-MPEG-1D-10X10 PRO-MPEG-1D-5X20 PRO-MPEG-2D-10X10 PRO-MPEG-2D-5X20)

name	*type*	*flags*	*range*
ipout.1.packets	CHOICE		EnumRange (TS ADVANCED-TS TTS)
ipout.1.tsinPacket	CHOICE		EnumRange (STANDARD MEDIUM LOW)
ipout.1.crypt	CHOICE		EnumRange (DISABLE CAMELLIA)
ipout.2.addressType	CHOICE		EnumRange (UNICAST MULTICAST)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (1024 .. 65530)
ipout.2.protocol	CHOICE		EnumRange (RTP UDP)
ipout.2.bitrate	INTEGER		IntegerRange (0 .. 0)
ipout.2.fecMode	CHOICE		EnumRange (DISABLE PRO-MPEG-1D-20X5 PRO-MPEG-1D-10X10 PRO-MPEG-1D-5X20 PRO-MPEG-2D-10X10 PRO-MPEG-2D-5X20)
ipout.2.packets	CHOICE		EnumRange (TS ADVANCED-TS TTS)
ipout.2.tsinPacket	CHOICE		EnumRange (STANDARD MEDIUM LOW)
ipout.2.crypt	CHOICE		EnumRange (DISABLE CAMELLIA)
ipout.2.outPort	CHOICE		EnumRange (DATA1 CONTROL-DATA2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video 1 input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Encode status)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Warning)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (FAN)

9.437 Narda-MCP

This page describes the driver configuration and the device window for the Narda-MCP device driver. This driver operates a Narda spectrum analyzer in multi channel power measurement mode. The power of up to twelve channels may be measured simultaneously, the measured values are compared against individual limits.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- Measurement Display (see below)
- --- Settings (see below)
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page.

Configuration parameters

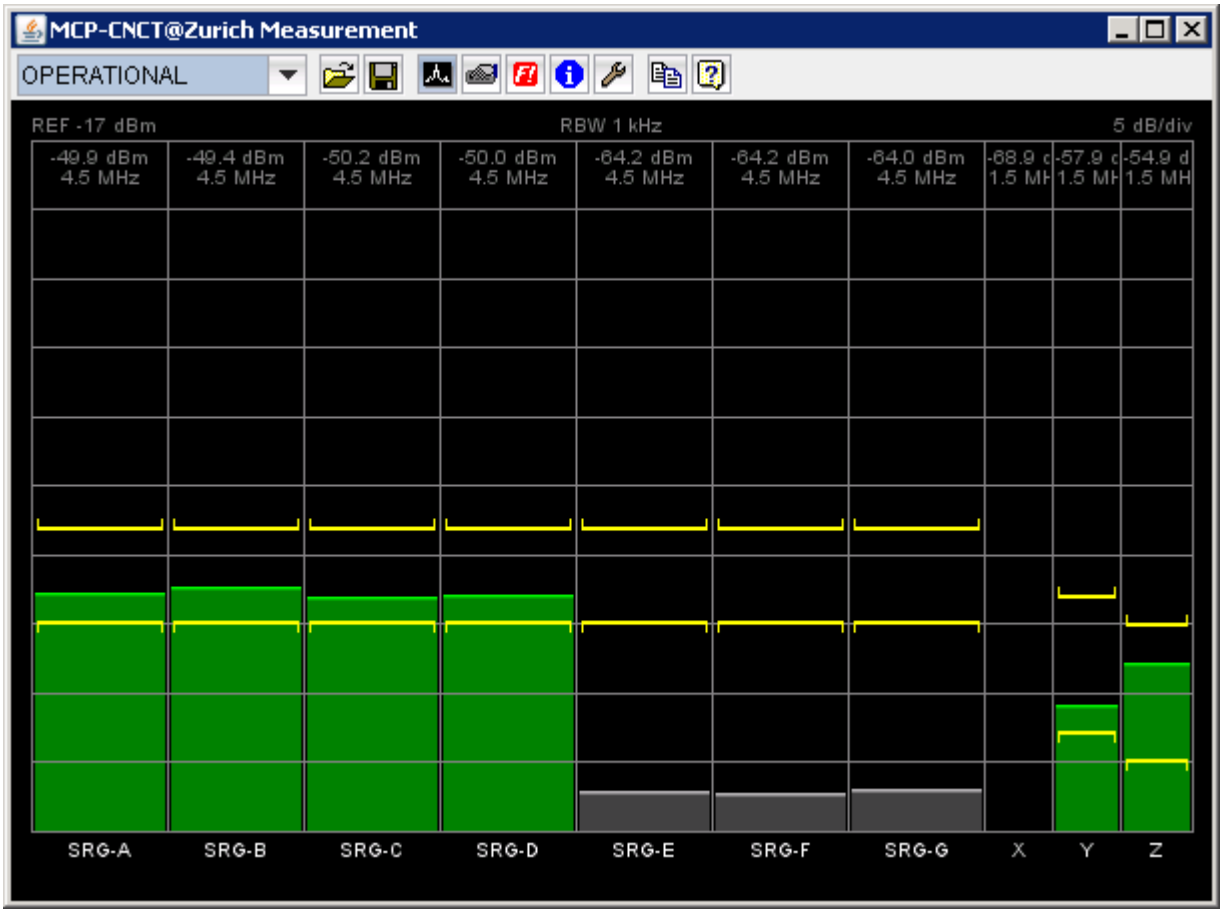
At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	You can leave this empty, the protocol does not use an address specification
loFrequency	The LO frequency which offsets the channel frequencies.

The Narda spectrum analyzer ist connected to the VLC as a TTY over TCP device, the IP address is specified in the device setup window as the interface name. The default opert number of the Narda spectrum analyzer is 5000.

Measurement display

The Narda-MCP device provides it's own graphical widget presenting the measured channel power values similar to a spectrum display. Each channel power value is shown as a colored bar with it's height telling the power value, it's width reflecting the channel's bandwidth (the width of a channel ist not always truely proportional to the channel bandwidth in order to prevent narrow bandwidth channels to collapse to a single line).



Channels with a power value within the given limits are drawn green, channels below or above their limits are red. Additionally a "noise level" value may be specified, channels below this value are considered to be off an are drawn gray (e.g. for occasional use transmissions where a not present is not treated as a fault.

Settings

At the settings page of the device window, the parameters of the up to twelve channels to measure are defined. For each channel, it's name, the center frequency, the bandwidth and the channel power limit values are set. All values are stored at the sat-nms VLC, not at the device itself.

Beside the channel parameter the "spectrum analyzer like" parameters *reference level*, *scale* and *resolution bandwidth* are defined here as well.

Finally, the special parameter *noise level* is used to set the display behaviour for channels with very low power. Channels with a power value below the noise level are considered to be switched off explicitly. No "low limit" fault is generated and the channel bar is drawn gray. This way you can differentiate between a "switched on, but power too low" and a "switched off, power doesn't care" condition. If you want a channel with low level to be reported as fault< in any case, set the noise level to a value like -999.9 dB.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.prodName	TEXT	R/O	StringRange R/O
info.prodID	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.deviceID	TEXT	R/O	StringRange R/O
info.fwRevision	TEXT	R/O	StringRange R/O
info.fwDate	TEXT	R/O	StringRange R/O
config.loFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
internal.dummy	TEXT	R/O	StringRange R/O
internal.sweeps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.swmem	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reflev	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
scale	CHOICE	SAVE	EnumRange (10 dB/div 5 dB/div 2 dB/div 1 dB/div)
rbw	FLOAT	SAVE NOCOMPARE	DoubleRange (0 .. 0)
noiselev	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.1.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.1.name	TEXT	SAVE	StringRange
chan.1.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
chan.1.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.1.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.1.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.1.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.1.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.1.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.2.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.2.name	TEXT	SAVE	StringRange
chan.2.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.2.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.2.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.2.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.2.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.2.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.2.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.3.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.3.name	TEXT	SAVE	StringRange
chan.3.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.3.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.3.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.3.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.3.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.3.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.3.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.4.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.4.name	TEXT	SAVE	StringRange
chan.4.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
chan.4.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.4.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.4.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.4.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.4.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.4.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.5.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.5.name	TEXT	SAVE	StringRange
chan.5.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.5.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.5.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.5.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.5.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.5.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.5.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.6.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.6.name	TEXT	SAVE	StringRange
chan.6.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.6.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.6.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.6.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.6.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.6.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.6.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.7.enable	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
chan.7.name	TEXT	SAVE	StringRange
chan.7.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.7.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.7.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.7.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.7.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.7.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.7.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.8.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.8.name	TEXT	SAVE	StringRange
chan.8.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.8.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.8.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.8.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.8.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.8.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.8.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.9.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.9.name	TEXT	SAVE	StringRange
chan.9.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.9.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.9.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.9.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.9.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.9.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.9.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.10.enable	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
chan.10.name	TEXT	SAVE	StringRange
chan.10.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.10.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.10.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.10.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.10.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.10.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.10.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.11.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.11.name	TEXT	SAVE	StringRange
chan.11.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.11.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.11.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.11.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.11.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.11.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.11.low	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.12.enable	CHOICE	SAVE	EnumRange (OFF ON)
chan.12.name	TEXT	SAVE	StringRange
chan.12.center	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.12.bw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
chan.12.loLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.12.hiLimit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
chan.12.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
chan.12.high	BOOLEAN	R/O	BooleanRange R/O (true false)
chan.12.low	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel 1 LOW)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Channel 1 HIGH)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel 2 LOW)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Channel 2 HIGH)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Channel 3 LOW)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Channel 3 HIGH)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Channel 4 LOW)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Channel 4 HIGH)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Channel 5 LOW)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Channel 5 HIGH)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Channel 6 LOW)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Channel 6 HIGH)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Channel 7 LOW)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Channel 7 HIGH)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Channel 8 LOW)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Channel 8 HIGH)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Channel 9 LOW)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Channel 9 HIGH)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Channel 10 LOW)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Channel 10 HIGH)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Channel 11 LOW)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Channel 11 HIGH)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Channel 12 LOW)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Channel 12 HIGH)

9.438 Narda-NRA

This page describes the *Narda NRA Series* device driver and the device window for this device.

The drivers was tested with following equipments:

- Model: NRA-3000
- SW version: V2.0.0 beta 7

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

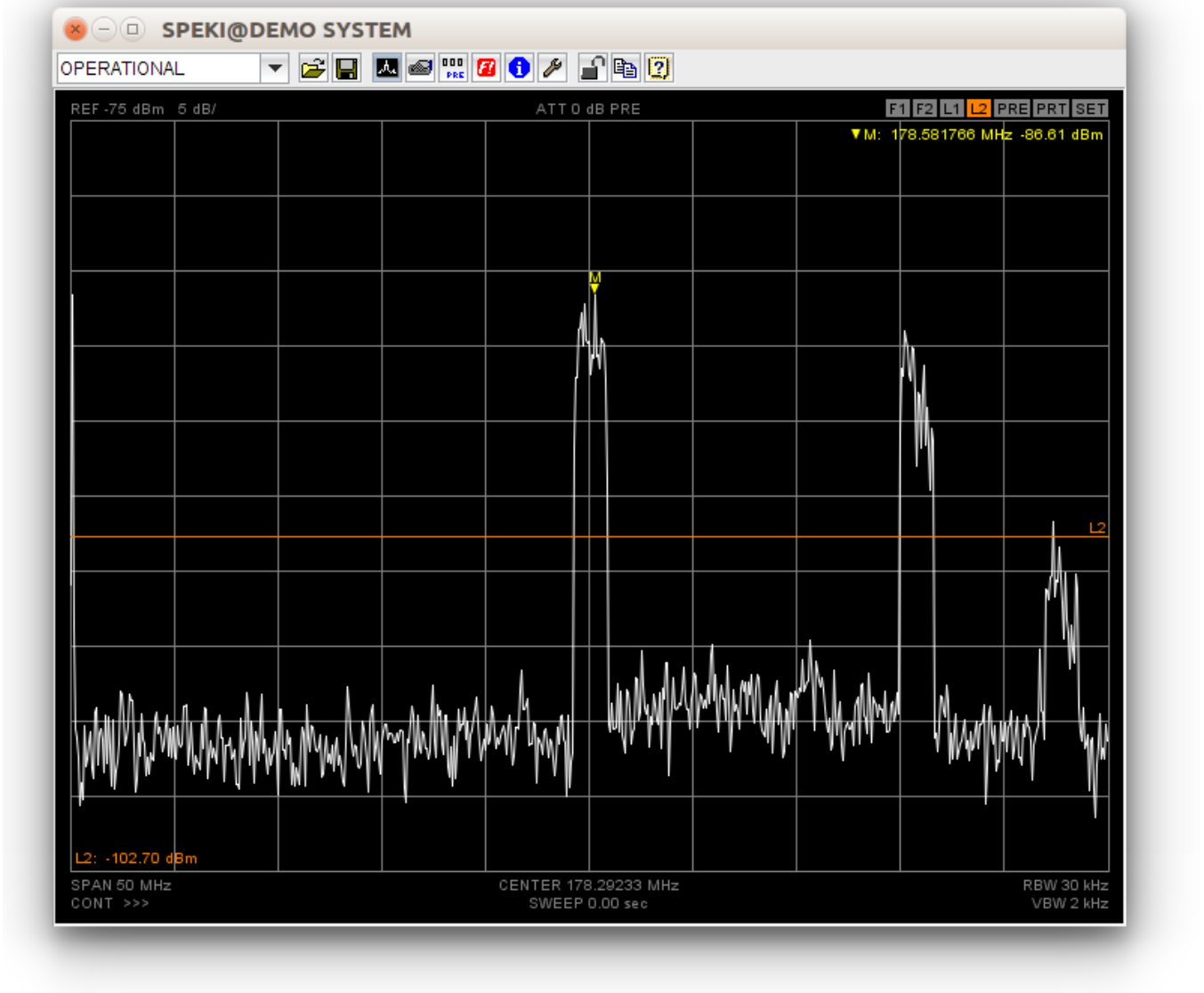
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page

- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values

which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Narda NRA driver does not offer a function for saving and loading presets.

How to configure the driver

The Narda NRA are controlled via a TCP connection at port 55555. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 55555. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwrevision	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange DIS (0 .. 65)
atten.auto	CHOICE		EnumRange DIS (AUTO FIX)
preamp	CHOICE		EnumRange DIS (OFF)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE	R/O	EnumRange R/O (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)

name	*type*	*flags*	*range*
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange DIS
recall	TEXT	NOPRESET	StringRange DIS

name	*type*	*flags*	*range*
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.439 Narda-Upconverter

This page describes the device driver and the device window for the Narda BUC/LKUX-1 upconverters. The driver has been tested with the L-band to Ku-band version of this upvonverter, but should work with other frequency ranges as well, as the limits of frequency and gain are read from the device.

The driver uses the binary packet protocol over RS485 to communicate with the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal 1..15) must match the value set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.spectrum	TEXT	R/O	StringRange R/O
info.unitType	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (17300.000 .. 18100.000)
internal.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.gmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.gmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fint	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.ffra	FLOAT	R/O	DoubleRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Internal comm.)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Checksum)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TX PLL lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General)

9.440 Negate

This page describes the device driver and the device window for the*Negate* logical device. The Negate logical device subscribes for a numeric value and evaluates the binary negation of this value. Negate may be used to invert the bit sense of a block of input signals monitored by a SatService IO-FEP or a third party PLC.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
inputId	The parameter ID of the PLC input block variable to negate. Example: **IOFEP.inputs.0** .
initializedId	If the Negate device is used in the sat-nms I/O port framework, Input or WG-Switch devices treat the Negate device a if this was a PLC or IO-FEP. Device drivers like the SatService-IO-FEP provide a status parameter called 'initialized' which becomes 'true' when all inputs have been read from the device once. You have to set the ID of this parameter here in order to make the Negate device replicate this initialized state. Example: **IOFEP.initialized** .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.input	HEX	R/O	HexRange R/O (0 .. 0)
info.output	HEX	R/O	HexRange R/O (0 .. 0)
config.inputId	TEXT	SAVE SETUP	StringRange
config.initializedId	TEXT	SAVE SETUP	StringRange
value	INTEGER	R/O	IntegerRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O

9.441 Netgear-M4300-Monitor

The *Netgear-M4300-Monitor* device driver monitors the state of up to 48 interfaces of a Netgear M4300 switch via SNMP. The driver is based on the MIB-2 interface monitor device driver. In addition to a textual description of the operational state (ifOperStatus) of each interface monitored, the driver provides fault signals for each interface not being 'up' and also the RX data rate, TX data rate and load interval. The load interval gives an indication of the integration time for the displayed data rate. This device driver is monitoring only.

The driver permits to configure the list of interfaces to be monitored. For this, the table index of each interface to monitor must be entered at the maintenance/setup page of the device window. As a side effect, devices which are numbering their interfaces non-sequentially are supported as well.

It is maybe necessary to enable the SNMP monitoring on the network switch via the Webinterface. **Device window pages**

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page whows the operational state of each interface monitored.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the redundancy switch in the sat-nms software.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read

and write community.

- ifIndex.1 ... ifIndex.48 --- For each interface to be monitored, the table index of the interface in the mib-2.interfaces has to be entered here. You find the index values with the documentation of the router or switch to monitor or simply by "walking" through the interfaces list of the device by means of a SNMP MIB browser tool. Setting an index value to zero means not to monitor this value.

Fault handling

For each of the 48 monitorable interfaces, the device provides a fault flag which is set if this interface is not "up". You may suppress this behaviour by setting the appropriate fault flags to "OFF" or "INFO", depending on the type of signalling desired for each individual interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sysDescr	TEXT	R/O	StringRange R/O
info.sysName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.image1	TEXT	R/O	StringRange R/O
info.image2	TEXT	R/O	StringRange R/O
info.activeImage	TEXT	R/O	StringRange R/O
info.sysUpTime	TEXT	R/O	StringRange R/O
config.ifIndex.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.7	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.8	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.9	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.10	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.11	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.12	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.13	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.14	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.15	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.16	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.17	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.18	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.19	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.20	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.21	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.22	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.23	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.24	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.25	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.26	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.27	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.28	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.29	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.30	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.31	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.32	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.33	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.ifIndex.34	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.35	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.36	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.37	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.38	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.39	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.40	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.41	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.42	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.43	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.44	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.45	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.46	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.47	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ifIndex.48	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
cpuMemoryFree	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cpuMemoryAvailable	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
cpuTotalUtilization	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
boxTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ifDescr.1	TEXT	R/O	StringRange R/O DIS
ifDescr.2	TEXT	R/O	StringRange R/O DIS
ifDescr.3	TEXT	R/O	StringRange R/O DIS
ifDescr.4	TEXT	R/O	StringRange R/O DIS
ifDescr.5	TEXT	R/O	StringRange R/O DIS
ifDescr.6	TEXT	R/O	StringRange R/O DIS
ifDescr.7	TEXT	R/O	StringRange R/O DIS
ifDescr.8	TEXT	R/O	StringRange R/O DIS
ifDescr.9	TEXT	R/O	StringRange R/O DIS
ifDescr.10	TEXT	R/O	StringRange R/O DIS
ifDescr.11	TEXT	R/O	StringRange R/O DIS
ifDescr.12	TEXT	R/O	StringRange R/O DIS
ifDescr.13	TEXT	R/O	StringRange R/O DIS
ifDescr.14	TEXT	R/O	StringRange R/O DIS
ifDescr.15	TEXT	R/O	StringRange R/O DIS
ifDescr.16	TEXT	R/O	StringRange R/O DIS
ifDescr.17	TEXT	R/O	StringRange R/O DIS
ifDescr.18	TEXT	R/O	StringRange R/O DIS
ifDescr.19	TEXT	R/O	StringRange R/O DIS
ifDescr.20	TEXT	R/O	StringRange R/O DIS
ifDescr.21	TEXT	R/O	StringRange R/O DIS
ifDescr.22	TEXT	R/O	StringRange R/O DIS
ifDescr.23	TEXT	R/O	StringRange R/O DIS
ifDescr.24	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifDescr.25	TEXT	R/O	StringRange R/O DIS
ifDescr.26	TEXT	R/O	StringRange R/O DIS
ifDescr.27	TEXT	R/O	StringRange R/O DIS
ifDescr.28	TEXT	R/O	StringRange R/O DIS
ifDescr.29	TEXT	R/O	StringRange R/O DIS
ifDescr.30	TEXT	R/O	StringRange R/O DIS
ifDescr.31	TEXT	R/O	StringRange R/O DIS
ifDescr.32	TEXT	R/O	StringRange R/O DIS
ifDescr.33	TEXT	R/O	StringRange R/O DIS
ifDescr.34	TEXT	R/O	StringRange R/O DIS
ifDescr.35	TEXT	R/O	StringRange R/O DIS
ifDescr.36	TEXT	R/O	StringRange R/O DIS
ifDescr.37	TEXT	R/O	StringRange R/O DIS
ifDescr.38	TEXT	R/O	StringRange R/O DIS
ifDescr.39	TEXT	R/O	StringRange R/O DIS
ifDescr.40	TEXT	R/O	StringRange R/O DIS
ifDescr.41	TEXT	R/O	StringRange R/O DIS
ifDescr.42	TEXT	R/O	StringRange R/O DIS
ifDescr.43	TEXT	R/O	StringRange R/O DIS
ifDescr.44	TEXT	R/O	StringRange R/O DIS
ifDescr.45	TEXT	R/O	StringRange R/O DIS
ifDescr.46	TEXT	R/O	StringRange R/O DIS
ifDescr.47	TEXT	R/O	StringRange R/O DIS
ifDescr.48	TEXT	R/O	StringRange R/O DIS
ifOperStatus.1	TEXT	R/O	StringRange R/O DIS
ifOperStatus.2	TEXT	R/O	StringRange R/O DIS
ifOperStatus.3	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifOperStatus.4	TEXT	R/O	StringRange R/O DIS
ifOperStatus.5	TEXT	R/O	StringRange R/O DIS
ifOperStatus.6	TEXT	R/O	StringRange R/O DIS
ifOperStatus.7	TEXT	R/O	StringRange R/O DIS
ifOperStatus.8	TEXT	R/O	StringRange R/O DIS
ifOperStatus.9	TEXT	R/O	StringRange R/O DIS
ifOperStatus.10	TEXT	R/O	StringRange R/O DIS
ifOperStatus.11	TEXT	R/O	StringRange R/O DIS
ifOperStatus.12	TEXT	R/O	StringRange R/O DIS
ifOperStatus.13	TEXT	R/O	StringRange R/O DIS
ifOperStatus.14	TEXT	R/O	StringRange R/O DIS
ifOperStatus.15	TEXT	R/O	StringRange R/O DIS
ifOperStatus.16	TEXT	R/O	StringRange R/O DIS
ifOperStatus.17	TEXT	R/O	StringRange R/O DIS
ifOperStatus.18	TEXT	R/O	StringRange R/O DIS
ifOperStatus.19	TEXT	R/O	StringRange R/O DIS
ifOperStatus.20	TEXT	R/O	StringRange R/O DIS
ifOperStatus.21	TEXT	R/O	StringRange R/O DIS
ifOperStatus.22	TEXT	R/O	StringRange R/O DIS
ifOperStatus.23	TEXT	R/O	StringRange R/O DIS
ifOperStatus.24	TEXT	R/O	StringRange R/O DIS
ifOperStatus.25	TEXT	R/O	StringRange R/O DIS
ifOperStatus.26	TEXT	R/O	StringRange R/O DIS
ifOperStatus.27	TEXT	R/O	StringRange R/O DIS
ifOperStatus.28	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ifOperStatus.29	TEXT	R/O	StringRange R/O DIS
ifOperStatus.30	TEXT	R/O	StringRange R/O DIS
ifOperStatus.31	TEXT	R/O	StringRange R/O DIS
ifOperStatus.32	TEXT	R/O	StringRange R/O DIS
ifOperStatus.33	TEXT	R/O	StringRange R/O DIS
ifOperStatus.34	TEXT	R/O	StringRange R/O DIS
ifOperStatus.35	TEXT	R/O	StringRange R/O DIS
ifOperStatus.36	TEXT	R/O	StringRange R/O DIS
ifOperStatus.37	TEXT	R/O	StringRange R/O DIS
ifOperStatus.38	TEXT	R/O	StringRange R/O DIS
ifOperStatus.39	TEXT	R/O	StringRange R/O DIS
ifOperStatus.40	TEXT	R/O	StringRange R/O DIS
ifOperStatus.41	TEXT	R/O	StringRange R/O DIS
ifOperStatus.42	TEXT	R/O	StringRange R/O DIS
ifOperStatus.43	TEXT	R/O	StringRange R/O DIS
ifOperStatus.44	TEXT	R/O	StringRange R/O DIS
ifOperStatus.45	TEXT	R/O	StringRange R/O DIS
ifOperStatus.46	TEXT	R/O	StringRange R/O DIS
ifOperStatus.47	TEXT	R/O	StringRange R/O DIS
ifOperStatus.48	TEXT	R/O	StringRange R/O DIS
ifRxRate.1	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.2	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.3	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.4	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifRxRate.5	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.6	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.7	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.8	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.9	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.10	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.11	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.12	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.13	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.14	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.15	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.16	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.17	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.18	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.19	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.20	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifRxRate.21	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.22	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.23	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.24	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.25	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.26	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.27	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.28	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.29	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.30	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.31	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.32	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.33	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.34	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.35	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.36	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifRxRate.37	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.38	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.39	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.40	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.41	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.42	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.43	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.44	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.45	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.46	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.47	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifRxRate.48	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.1	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.2	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.3	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.4	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifTxRate.5	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.6	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.7	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.8	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.9	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.10	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.11	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.12	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.13	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.14	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.15	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.16	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.17	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.18	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.19	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.20	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifTxRate.21	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.22	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.23	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.24	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.25	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.26	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.27	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.28	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.29	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.30	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.31	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.32	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.33	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.34	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.35	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.36	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)

name	*type*	*flags*	*range*
ifTxRate.37	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.38	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.39	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.40	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.41	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.42	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.43	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.44	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.45	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.46	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.47	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifTxRate.48	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 10000.000)
ifLoadInterval.1	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.2	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.3	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.4	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.5	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.6	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.7	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)

name	*type*	*flags*	*range*
ifLoadInterval.8	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.9	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.10	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.11	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.12	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.13	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.14	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.15	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.16	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.17	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.18	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.19	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.20	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.21	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.22	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.23	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.24	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.25	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.26	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.27	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.28	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.29	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.30	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.31	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.32	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.33	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)

name	*type*	*flags*	*range*
ifLoadInterval.34	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.35	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.36	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.37	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.38	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.39	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.40	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.41	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.42	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.43	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.44	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.45	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.46	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.47	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifLoadInterval.48	INTEGER	R/O	IntegerRange R/O DIS (30 .. 600)
ifComment.1	TEXT	SAVE	StringRange
ifComment.2	TEXT	SAVE	StringRange
ifComment.3	TEXT	SAVE	StringRange
ifComment.4	TEXT	SAVE	StringRange
ifComment.5	TEXT	SAVE	StringRange
ifComment.6	TEXT	SAVE	StringRange
ifComment.7	TEXT	SAVE	StringRange
ifComment.8	TEXT	SAVE	StringRange
ifComment.9	TEXT	SAVE	StringRange
ifComment.10	TEXT	SAVE	StringRange
ifComment.11	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
ifComment.12	TEXT	SAVE	StringRange
ifComment.13	TEXT	SAVE	StringRange
ifComment.14	TEXT	SAVE	StringRange
ifComment.15	TEXT	SAVE	StringRange
ifComment.16	TEXT	SAVE	StringRange
ifComment.17	TEXT	SAVE	StringRange
ifComment.18	TEXT	SAVE	StringRange
ifComment.19	TEXT	SAVE	StringRange
ifComment.20	TEXT	SAVE	StringRange
ifComment.21	TEXT	SAVE	StringRange
ifComment.22	TEXT	SAVE	StringRange
ifComment.23	TEXT	SAVE	StringRange
ifComment.24	TEXT	SAVE	StringRange
ifComment.25	TEXT	SAVE	StringRange
ifComment.26	TEXT	SAVE	StringRange
ifComment.27	TEXT	SAVE	StringRange
ifComment.28	TEXT	SAVE	StringRange
ifComment.29	TEXT	SAVE	StringRange
ifComment.30	TEXT	SAVE	StringRange
ifComment.31	TEXT	SAVE	StringRange
ifComment.32	TEXT	SAVE	StringRange
ifComment.33	TEXT	SAVE	StringRange
ifComment.34	TEXT	SAVE	StringRange
ifComment.35	TEXT	SAVE	StringRange
ifComment.36	TEXT	SAVE	StringRange
ifComment.37	TEXT	SAVE	StringRange
ifComment.38	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
ifComment.39	TEXT	SAVE	StringRange
ifComment.40	TEXT	SAVE	StringRange
ifComment.41	TEXT	SAVE	StringRange
ifComment.42	TEXT	SAVE	StringRange
ifComment.43	TEXT	SAVE	StringRange
ifComment.44	TEXT	SAVE	StringRange
ifComment.45	TEXT	SAVE	StringRange
ifComment.46	TEXT	SAVE	StringRange
ifComment.47	TEXT	SAVE	StringRange
ifComment.48	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Ifc01 status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ifc02 status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Ifc03 status)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ifc04 status)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ifc05 status)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Ifc06 status)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Ifc07 status)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Ifc08 status)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Ifc09 status)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ifc10 status)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Ifc11 status)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ifc12 status)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Ifc13 status)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Ifc14 status)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Ifc15 status)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Ifc16 status)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Ifc17 status)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Ifc18 status)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Ifc19 status)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Ifc20 status)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Ifc21 status)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Ifc22 status)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Ifc23 status)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Ifc24 status)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Ifc25 status)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Ifc26 status)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Ifc27 status)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Ifc28 status)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Ifc29 status)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Ifc30 status)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Ifc31 status)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Ifc32 status)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Ifc33 status)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Ifc34 status)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Ifc35 status)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Ifc36 status)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Ifc37 status)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Ifc38 status)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Ifc39 status)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Ifc40 status)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Ifc41 status)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (Ifc42 status)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Ifc43 status)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Ifc44 status)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Ifc45 status)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Ifc46 status)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Ifc47 status)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Ifc48 status)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Temp. Sensor 1)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Temp. Sensor 2)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Temperature Unit)

9.442 Nevion-Sublime

This page describes the device driver and the device window for a Nevion ASI Switch Matrix (Sublime series) with upto 16 inputs and 2 outputs. The driver supports only interfaces on Level 0.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Matrix do not need any configuration parameters except the settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device

Remarks

- internal port numbers of the Nevion protocol starts with 0
- requires syntax v128
- turn crc off.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.level1	TEXT	R/O	StringRange R/O
info.login	TEXT	R/O	StringRange R/O
info.syntax	TEXT	R/O	StringRange R/O
info.crc	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
output.02	CHOICE		EnumRange (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.443 Nevion-TVG420

This page describes the device driver and the device window for the ASI to IP and IP to ASI Gateway Nevion TVG 420 via SNMP.

The driver allows to change each of the 4 stream interfaces direction from ASI to IP or IP to ASI and vice versa. The respective parameters are en/disabled accordingly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Stream 1 direction and parameters.
- --- Stream 2 direction and parameters
- --- Stream 3 direction and parameters
- --- Stream 4 direction and parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If

both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
ip.01.direction	CHOICE		EnumRange (ASI TO IP IP TO ASI)
ip.02.direction	CHOICE		EnumRange (ASI TO IP IP TO ASI)
ip.03.direction	CHOICE		EnumRange (ASI TO IP IP TO ASI)
ip.04.direction	CHOICE		EnumRange (ASI TO IP IP TO ASI)
asiin.01.enable	CHOICE		EnumRange (OFF ON)
asiin.02.enable	CHOICE		EnumRange (OFF ON)
asiin.03.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
asiin.04.enable	CHOICE		EnumRange (OFF ON)
ipout.01.name	TEXT		StringRange
ipout.02.name	TEXT		StringRange
ipout.03.name	TEXT		StringRange
ipout.04.name	TEXT		StringRange
ipout.01.enable	CHOICE		EnumRange (OFF ON)
ipout.02.enable	CHOICE		EnumRange (OFF ON)
ipout.03.enable	CHOICE		EnumRange (OFF ON)
ipout.04.enable	CHOICE		EnumRange (OFF ON)
ipout.01.address	TEXT		StringRange
ipout.02.address	TEXT		StringRange
ipout.03.address	TEXT		StringRange
ipout.04.address	TEXT		StringRange
ipout.01.port	INTEGER		IntegerRange (0 .. 65535)
ipout.02.port	INTEGER		IntegerRange (0 .. 65535)
ipout.03.port	INTEGER		IntegerRange (0 .. 65535)
ipout.04.port	INTEGER		IntegerRange (0 .. 65535)
ipout.01.srcAddress	TEXT	R/O	StringRange R/O
ipout.02.srcAddress	TEXT	R/O	StringRange R/O
ipout.03.srcAddress	TEXT	R/O	StringRange R/O
ipout.04.srcAddress	TEXT	R/O	StringRange R/O
ipout.01.srcPort	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
ipout.02.srcPort	INTEGER		IntegerRange (0 .. 65535)
ipout.03.srcPort	INTEGER		IntegerRange (0 .. 65535)
ipout.04.srcPort	INTEGER		IntegerRange (0 .. 65535)
ipout.01.ttl	INTEGER		IntegerRange (0 .. 255)
ipout.02.ttl	INTEGER		IntegerRange (0 .. 255)
ipout.03.ttl	INTEGER		IntegerRange (0 .. 255)
ipout.04.ttl	INTEGER		IntegerRange (0 .. 255)
ipout.01.packetsFrame	INTEGER		IntegerRange (1 .. 7)
ipout.02.packetsFrame	INTEGER		IntegerRange (1 .. 7)
ipout.03.packetsFrame	INTEGER		IntegerRange (1 .. 7)
ipout.04.packetsFrame	INTEGER		IntegerRange (1 .. 7)
ipout.01.tos	INTEGER		IntegerRange (0 .. 255)
ipout.02.tos	INTEGER		IntegerRange (0 .. 255)
ipout.03.tos	INTEGER		IntegerRange (0 .. 255)
ipout.04.tos	INTEGER		IntegerRange (0 .. 255)
ipout.01.routerEnable	CHOICE		EnumRange (OFF ON)
ipout.02.routerEnable	CHOICE		EnumRange (OFF ON)
ipout.03.routerEnable	CHOICE		EnumRange (OFF ON)
ipout.04.routerEnable	CHOICE		EnumRange (OFF ON)
asiin.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
asiin.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.01.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.02.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.03.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.04.packetSize	CHOICE	R/O	EnumRange R/O (188 204)
asiin.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.01.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.02.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.03.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiin.04.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.01.addressResolved	BOOLEAN	R/O	BooleanRange R/O (true false)
ipout.02.addressResolved	BOOLEAN	R/O	BooleanRange R/O (true false)
ipout.03.addressResolved	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ipout.04.addressResolved	BOOLEAN	R/O	BooleanRange R/O (true false)
ipout.01.Mac	TEXT	R/O	StringRange R/O
ipout.02.Mac	TEXT	R/O	StringRange R/O
ipout.03.Mac	TEXT	R/O	StringRange R/O
ipout.04.Mac	TEXT	R/O	StringRange R/O
ipout.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.01.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.02.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.03.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.04.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.01.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.02.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.03.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)

name	*type*	*flags*	*range*
ipout.04.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.01.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.02.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.03.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.04.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.01.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.02.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.03.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.04.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.01.fecColumnsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.02.fecColumnsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.03.fecColumnsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.04.fecColumnsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.01.fecRowsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.02.fecRowsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.03.fecRowsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.04.fecRowsDatarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.01.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.02.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)

name	*type*	*flags*	*range*
ipout.03.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.04.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.01.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.02.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.03.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.04.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipout.01.vLanEnable	CHOICE		EnumRange (OFF ON)
ipout.02.vLanEnable	CHOICE		EnumRange (OFF ON)
ipout.03.vLanEnable	CHOICE		EnumRange (OFF ON)
ipout.04.vLanEnable	CHOICE		EnumRange (OFF ON)
ipout.01.vLanTag	INTEGER		IntegerRange (0 .. 0)
ipout.02.vLanTag	INTEGER		IntegerRange (0 .. 0)
ipout.03.vLanTag	INTEGER		IntegerRange (0 .. 0)
ipout.04.vLanTag	INTEGER		IntegerRange (0 .. 0)
ipout.01.vLanPriority	INTEGER		IntegerRange (0 .. 0)
ipout.02.vLanPriority	INTEGER		IntegerRange (0 .. 0)
ipout.03.vLanPriority	INTEGER		IntegerRange (0 .. 0)
ipout.04.vLanPriority	INTEGER		IntegerRange (0 .. 0)
ipout.01.framesDropRate	FLOAT		DoubleRange (0 .. 100)
ipout.02.framesDropRate	FLOAT		DoubleRange (0 .. 100)
ipout.03.framesDropRate	FLOAT		DoubleRange (0 .. 100)
ipout.04.framesDropRate	FLOAT		DoubleRange (0 .. 100)
ipout.01.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipout.02.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.03.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.04.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.01.fecSkew	CHOICE		EnumRange (OFF ON)
ipout.02.fecSkew	CHOICE		EnumRange (OFF ON)
ipout.03.fecSkew	CHOICE		EnumRange (OFF ON)
ipout.04.fecSkew	CHOICE		EnumRange (OFF ON)
ipout.01.protocol	CHOICE		EnumRange (RTP UDP)
ipout.02.protocol	CHOICE		EnumRange (RTP UDP)
ipout.03.protocol	CHOICE		EnumRange (RTP UDP)
ipout.04.protocol	CHOICE		EnumRange (RTP UDP)
asiin.01.keep204	CHOICE		EnumRange (OFF ON)
asiin.02.keep204	CHOICE		EnumRange (OFF ON)
asiin.03.keep204	CHOICE		EnumRange (OFF ON)
asiin.04.keep204	CHOICE		EnumRange (OFF ON)
asiin.01.maxBitrate	INTEGER		IntegerRange (0 .. 210)
asiin.02.maxBitrate	INTEGER		IntegerRange (0 .. 210)
asiin.03.maxBitrate	INTEGER		IntegerRange (0 .. 210)
asiin.04.maxBitrate	INTEGER		IntegerRange (0 .. 210)
ipout.01.staticMac.on	CHOICE		EnumRange (OFF ON)
ipout.02.staticMac.on	CHOICE		EnumRange (OFF ON)
ipout.03.staticMac.on	CHOICE		EnumRange (OFF ON)
ipout.04.staticMac.on	CHOICE		EnumRange (OFF ON)
ipout.01.staticMac	TEXT		StringRange
ipout.02.staticMac	TEXT		StringRange
ipout.03.staticMac	TEXT		StringRange
ipout.04.staticMac	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.01.spoofing	CHOICE		EnumRange (OFF ON)
ipout.02.spoofing	CHOICE		EnumRange (OFF ON)
ipout.03.spoofing	CHOICE		EnumRange (OFF ON)
ipout.04.spoofing	CHOICE		EnumRange (OFF ON)
ipout.01.spoofingAddress	TEXT		StringRange
ipout.02.spoofingAddress	TEXT		StringRange
ipout.03.spoofingAddress	TEXT		StringRange
ipout.04.spoofingAddress	TEXT		StringRange
ipout.01.interface	CHOICE		EnumRange (CONTROL DATA)
ipout.02.interface	CHOICE		EnumRange (CONTROL DATA)
ipout.03.interface	CHOICE		EnumRange (CONTROL DATA)
ipout.04.interface	CHOICE		EnumRange (CONTROL DATA)
asiout.01.enable	CHOICE		EnumRange (OFF ON)
asiout.02.enable	CHOICE		EnumRange (OFF ON)
asiout.03.enable	CHOICE		EnumRange (OFF ON)
asiout.04.enable	CHOICE		EnumRange (OFF ON)
ipin.01.name	TEXT		StringRange
ipin.02.name	TEXT		StringRange
ipin.03.name	TEXT		StringRange
ipin.04.name	TEXT		StringRange
ipin.01.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.02.mode	CHOICE		EnumRange (UNICAST MULTICAST)

name	*type*	*flags*	*range*
ipin.03.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.04.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.01.address	TEXT		StringRange
ipin.02.address	TEXT		StringRange
ipin.03.address	TEXT		StringRange
ipin.04.address	TEXT		StringRange
ipin.01.port	INTEGER		IntegerRange (0 .. 65535)
ipin.02.port	INTEGER		IntegerRange (0 .. 65535)
ipin.03.port	INTEGER		IntegerRange (0 .. 65535)
ipin.04.port	INTEGER		IntegerRange (0 .. 65535)
ipin.01.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.02.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.03.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.04.bufferDuration	INTEGER		IntegerRange (0 .. 0)
ipin.01.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ipin.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiout.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.01.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.02.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.03.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
asiout.04.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipin.01.bufferFillSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.bufferFillSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.bufferFillSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.bufferFillSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.02.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.03.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.04.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.01.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.01.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.02.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.03.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.04.fecColumnsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.01.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.02.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)

name	*type*	*flags*	*range*
ipin.03.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.04.fecRowsPort	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
ipin.01.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.fecColumnsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.fecColumnsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.fecColumnsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.fecColumnsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.fecRowsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.fecRowsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.fecRowsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.fecRowsSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.reset	TEXT	NOPRESET	StringRange
ipin.02.reset	TEXT	NOPRESET	StringRange
ipin.03.reset	TEXT	NOPRESET	StringRange
ipin.04.reset	TEXT	NOPRESET	StringRange
ipin.01.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.srcAddress	TEXT		StringRange
ipin.02.srcAddress	TEXT		StringRange
ipin.03.srcAddress	TEXT		StringRange

name	*type*	*flags*	*range*
ipin.04.srcAddress	TEXT		StringRange
ipin.01.measLatency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipin.02.measLatency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipin.03.measLatency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipin.04.measLatency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipin.01.latency	FLOAT		DoubleRange (0.0 .. 0.0)
ipin.02.latency	FLOAT		DoubleRange (0.0 .. 0.0)
ipin.03.latency	FLOAT		DoubleRange (0.0 .. 0.0)
ipin.04.latency	FLOAT		DoubleRange (0.0 .. 0.0)
ipin.01.maxDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.maxDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.maxDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.maxDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.fail	CHOICE		EnumRange (OFF IDLE)
ipin.02.fail	CHOICE		EnumRange (OFF IDLE)
ipin.03.fail	CHOICE		EnumRange (OFF IDLE)
ipin.04.fail	CHOICE		EnumRange (OFF IDLE)
ipin.01.asiMode	CHOICE		EnumRange (SPREAD BURST)

name	*type*	*flags*	*range*
ipin.02.asiMode	CHOICE		EnumRange (SPREAD BURST)
ipin.03.asiMode	CHOICE		EnumRange (SPREAD BURST)
ipin.04.asiMode	CHOICE		EnumRange (SPREAD BURST)
ipin.01.bitrateMode	CHOICE		EnumRange (PCR COARSE MIP VBR)
ipin.02.bitrateMode	CHOICE		EnumRange (PCR COARSE MIP VBR)
ipin.03.bitrateMode	CHOICE		EnumRange (PCR COARSE MIP VBR)
ipin.04.bitrateMode	CHOICE		EnumRange (PCR COARSE MIP VBR)
ipin.01.bitrateMonitoring	CHOICE		EnumRange (OFF ON)
ipin.02.bitrateMonitoring	CHOICE		EnumRange (OFF ON)
ipin.03.bitrateMonitoring	CHOICE		EnumRange (OFF ON)
ipin.04.bitrateMonitoring	CHOICE		EnumRange (OFF ON)
ipin.01.rateMax	INTEGER		IntegerRange (0 .. 0)
ipin.02.rateMax	INTEGER		IntegerRange (0 .. 0)
ipin.03.rateMax	INTEGER		IntegerRange (0 .. 0)
ipin.04.rateMax	INTEGER		IntegerRange (0 .. 0)
ipin.01.nullPackets	CHOICE		EnumRange (OFF ON)
ipin.02.nullPackets	CHOICE		EnumRange (OFF ON)
ipin.03.nullPackets	CHOICE		EnumRange (OFF ON)
ipin.04.nullPackets	CHOICE		EnumRange (OFF ON)
ipin.01.meas.rateMode	TEXT	R/O	StringRange R/O
ipin.02.meas.rateMode	TEXT	R/O	StringRange R/O
ipin.03.meas.rateMode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ipin.04.meas.rateMode	TEXT	R/O	StringRange R/O
ipin.01.currentBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.02.currentBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.03.currentBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.04.currentBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.01.initialBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.02.initialBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.03.initialBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.04.initialBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.01.measBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.02.measBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.03.measBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.04.measBitrateRegulator	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.01.stateRegulator	TEXT	R/O	StringRange R/O
ipin.02.stateRegulator	TEXT	R/O	StringRange R/O
ipin.03.stateRegulator	TEXT	R/O	StringRange R/O
ipin.04.stateRegulator	TEXT	R/O	StringRange R/O
asiout.01.packetSize	CHOICE		EnumRange (188 204)
asiout.02.packetSize	CHOICE		EnumRange (188 204)

name	*type*	*flags*	*range*
asiout.03.packetSize	CHOICE		EnumRange (188 204)
asiout.04.packetSize	CHOICE		EnumRange (188 204)
ipin.01.packetSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.packetSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.packetSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.packetSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.interface	CHOICE		EnumRange (CONTROL DATA)
ipin.02.interface	CHOICE		EnumRange (CONTROL DATA)
ipin.03.interface	CHOICE		EnumRange (CONTROL DATA)
ipin.04.interface	CHOICE		EnumRange (CONTROL DATA)
faults.01	ALARM	R/O	AlarmFlagRange R/O (ASI 1 Sync)
faults.02	ALARM	R/O	AlarmFlagRange R/O (ASI 2 Sync)
faults.03	ALARM	R/O	AlarmFlagRange R/O (ASI 3 Sync)
faults.04	ALARM	R/O	AlarmFlagRange R/O (ASI 4 Sync)
faults.05	ALARM	R/O	AlarmFlagRange R/O (ASI 1 High Bitrate)
faults.06	ALARM	R/O	AlarmFlagRange R/O (ASI 2 High Bitrate)
faults.07	ALARM	R/O	AlarmFlagRange R/O (ASI 3 High Bitrate)
faults.08	ALARM	R/O	AlarmFlagRange R/O (ASI 4 High Bitrate)
faults.09	ALARM	R/O	AlarmFlagRange R/O (IP 1 Transmit)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (IP 2 Transmit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (IP 3 Transmit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (IP 4 Transmit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IP 1 Inp Regulator)
faults.14	ALARM	R/O	AlarmFlagRange R/O (IP 2 Inp Regulator)
faults.15	ALARM	R/O	AlarmFlagRange R/O (IP 3 Inp Regulator)
faults.16	ALARM	R/O	AlarmFlagRange R/O (IP 4 Inp Regulator)
faults.17	ALARM	R/O	AlarmFlagRange R/O (ETH Data Link)
faults.18	ALARM	R/O	AlarmFlagRange R/O (ETH Output Ovfl)
faults.19	ALARM	R/O	AlarmFlagRange R/O (IP 1 Address)
faults.20	ALARM	R/O	AlarmFlagRange R/O (IP 2 Address)
faults.21	ALARM	R/O	AlarmFlagRange R/O (IP 3 Address)
faults.22	ALARM	R/O	AlarmFlagRange R/O (IP 4 Address)
faults.23	ALARM	R/O	AlarmFlagRange R/O (IP 1 RTP Sequence)
faults.24	ALARM	R/O	AlarmFlagRange R/O (IP 2 RTP Sequence)
faults.25	ALARM	R/O	AlarmFlagRange R/O (IP 3 RTP Sequence)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (IP 4 RTP Sequence)
faults.27	ALARM	R/O	AlarmFlagRange R/O (IP 1 Data)
faults.28	ALARM	R/O	AlarmFlagRange R/O (IP 2 Data)
faults.29	ALARM	R/O	AlarmFlagRange R/O (IP 3 Data)
faults.30	ALARM	R/O	AlarmFlagRange R/O (IP 4 Data)
faults.31	ALARM	R/O	AlarmFlagRange R/O (IP 1 FEC Thresh)
faults.32	ALARM	R/O	AlarmFlagRange R/O (IP 2 FEC Thresh)
faults.33	ALARM	R/O	AlarmFlagRange R/O (IP 3 FEC Thresh)
faults.34	ALARM	R/O	AlarmFlagRange R/O (IP 4 FEC Thresh)
faults.35	ALARM	R/O	AlarmFlagRange R/O (IP 1 Data Lost)
faults.36	ALARM	R/O	AlarmFlagRange R/O (IP 2 Data Lost)
faults.37	ALARM	R/O	AlarmFlagRange R/O (IP 3 Data Lost)
faults.38	ALARM	R/O	AlarmFlagRange R/O (IP 4 Data Lost)
faults.39	ALARM	R/O	AlarmFlagRange R/O (IP 1 Lock)
faults.40	ALARM	R/O	AlarmFlagRange R/O (IP 2 Lock)
faults.41	ALARM	R/O	AlarmFlagRange R/O (IP 3 Lock)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (IP 4 Lock)
faults.43	ALARM	R/O	AlarmFlagRange R/O (IP 1 MIP Missing)
faults.44	ALARM	R/O	AlarmFlagRange R/O (IP 2 MIP Missing)
faults.45	ALARM	R/O	AlarmFlagRange R/O (IP 3 MIP Missing)
faults.46	ALARM	R/O	AlarmFlagRange R/O (IP 4 MIP Missing)
faults.47	ALARM	R/O	AlarmFlagRange R/O (IP 1 Latency)
faults.48	ALARM	R/O	AlarmFlagRange R/O (IP 2 Latency)
faults.49	ALARM	R/O	AlarmFlagRange R/O (IP 3 Latency)
faults.50	ALARM	R/O	AlarmFlagRange R/O (IP 4 Latency)
faults.51	ALARM	R/O	AlarmFlagRange R/O (IP Input Ovfl)
faults.52	ALARM	R/O	AlarmFlagRange R/O (SNTP)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Time Ref)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.57	ALARM	R/O	AlarmFlagRange R/O (GPS Sync)

name	*type*	*flags*	*range*
faults.58	ALARM	R/O	AlarmFlagRange R/O (Other Warnings)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Other Faults)

9.444 Nevion-TVG425

This page describes the device driver and the device window for the Nevion TVG-425 transport stream gateway.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Stream overview
- --- Stream 1 settings and state
- --- Stream 2 settings and state
- --- Stream 3 settings and state
- --- Stream 4 settings and state
- --- Stream 5 settings and state
- --- Stream 6 settings and state
- --- Stream 7 settings and state
- --- Stream 8 settings and state
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Remarks

- The driver does not permit to change the device's stream configuration, i.e. which ASI or IP inputs/outputs belong to which stream. It however reads this configuration from device every 5 seconds and assigns the ports to streams according to the configuration read.
- The driver addresses the ASI / IP inputs and outputs by stream, not by the physical port number. This in particular means that e.g. the *asiin.03.** variables refer to the ASI port actually defined as input for stream 3, not to the physical port number 3. You may read the variable *asiin.03.port* in this case to evaluate the number of the physical port the stream uses.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
i.updated	TEXT	R/O	StringRange R/O
i.01.ain	TEXT	R/O	StringRange R/O
i.02.ain	TEXT	R/O	StringRange R/O
i.03.ain	TEXT	R/O	StringRange R/O
i.04.ain	TEXT	R/O	StringRange R/O
i.05.ain	TEXT	R/O	StringRange R/O
i.06.ain	TEXT	R/O	StringRange R/O
i.07.ain	TEXT	R/O	StringRange R/O
i.08.ain	TEXT	R/O	StringRange R/O
i.01.ipin	TEXT	R/O	StringRange R/O
i.02.ipin	TEXT	R/O	StringRange R/O
i.03.ipin	TEXT	R/O	StringRange R/O
i.04.ipin	TEXT	R/O	StringRange R/O
i.05.ipin	TEXT	R/O	StringRange R/O
i.06.ipin	TEXT	R/O	StringRange R/O
i.07.ipin	TEXT	R/O	StringRange R/O
i.08.ipin	TEXT	R/O	StringRange R/O
i.01.ipinrx	TEXT	R/O	StringRange R/O
i.02.ipinrx	TEXT	R/O	StringRange R/O
i.03.ipinrx	TEXT	R/O	StringRange R/O
i.04.ipinrx	TEXT	R/O	StringRange R/O
i.05.ipinrx	TEXT	R/O	StringRange R/O
i.06.ipinrx	TEXT	R/O	StringRange R/O
i.07.ipinrx	TEXT	R/O	StringRange R/O
i.08.ipinrx	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
i.01.aout	TEXT	R/O	StringRange R/O
i.02.aout	TEXT	R/O	StringRange R/O
i.03.aout	TEXT	R/O	StringRange R/O
i.04.aout	TEXT	R/O	StringRange R/O
i.05.aout	TEXT	R/O	StringRange R/O
i.06.aout	TEXT	R/O	StringRange R/O
i.07.aout	TEXT	R/O	StringRange R/O
i.08.aout	TEXT	R/O	StringRange R/O
i.01.ipout	TEXT	R/O	StringRange R/O
i.02.ipout	TEXT	R/O	StringRange R/O
i.03.ipout	TEXT	R/O	StringRange R/O
i.04.ipout	TEXT	R/O	StringRange R/O
i.05.ipout	TEXT	R/O	StringRange R/O
i.06.ipout	TEXT	R/O	StringRange R/O
i.07.ipout	TEXT	R/O	StringRange R/O
i.08.ipout	TEXT	R/O	StringRange R/O
i.01.ipouttx	TEXT	R/O	StringRange R/O
i.02.ipouttx	TEXT	R/O	StringRange R/O
i.03.ipouttx	TEXT	R/O	StringRange R/O
i.04.ipouttx	TEXT	R/O	StringRange R/O
i.05.ipouttx	TEXT	R/O	StringRange R/O
i.06.ipouttx	TEXT	R/O	StringRange R/O
i.07.ipouttx	TEXT	R/O	StringRange R/O
i.08.ipouttx	TEXT	R/O	StringRange R/O
stream.01.name	TEXT	R/O	StringRange R/O
stream.02.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
stream.03.name	TEXT	R/O	StringRange R/O
stream.04.name	TEXT	R/O	StringRange R/O
stream.05.name	TEXT	R/O	StringRange R/O
stream.06.name	TEXT	R/O	StringRange R/O
stream.07.name	TEXT	R/O	StringRange R/O
stream.08.name	TEXT	R/O	StringRange R/O
stream.01.enable	CHOICE		EnumRange (OFF ON)
stream.02.enable	CHOICE		EnumRange (OFF ON)
stream.03.enable	CHOICE		EnumRange (OFF ON)
stream.04.enable	CHOICE		EnumRange (OFF ON)
stream.05.enable	CHOICE		EnumRange (OFF ON)
stream.06.enable	CHOICE		EnumRange (OFF ON)
stream.07.enable	CHOICE		EnumRange (OFF ON)
stream.08.enable	CHOICE		EnumRange (OFF ON)
stream.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.05.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.06.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.07.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.08.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.01.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.02.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.03.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.04.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
stream.05.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.06.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.07.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.08.effectiveBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
stream.01.input	CHOICE		EnumRange (0 OFF)
stream.02.input	CHOICE		EnumRange (0 OFF)
stream.03.input	CHOICE		EnumRange (0 OFF)
stream.04.input	CHOICE		EnumRange (0 OFF)
stream.05.input	CHOICE		EnumRange (0 OFF)
stream.06.input	CHOICE		EnumRange (0 OFF)
stream.07.input	CHOICE		EnumRange (0 OFF)
stream.08.input	CHOICE		EnumRange (0 OFF)
stream.01.imode	TEXT	R/O	StringRange R/O
stream.02.imode	TEXT	R/O	StringRange R/O
stream.03.imode	TEXT	R/O	StringRange R/O
stream.04.imode	TEXT	R/O	StringRange R/O
stream.05.imode	TEXT	R/O	StringRange R/O
stream.06.imode	TEXT	R/O	StringRange R/O
stream.07.imode	TEXT	R/O	StringRange R/O
stream.08.imode	TEXT	R/O	StringRange R/O
stream.01.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.02.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.03.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.04.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.05.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.06.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
stream.07.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.08.hasIpOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.01.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.02.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.03.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.04.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.05.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.06.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.07.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.08.hasAsiOut	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.01.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.02.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.03.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.04.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.05.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.06.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.07.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
stream.08.txEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.01.reset	TEXT	NOPRESET	StringRange
ipin.02.reset	TEXT	NOPRESET	StringRange
ipin.03.reset	TEXT	NOPRESET	StringRange
ipin.04.reset	TEXT	NOPRESET	StringRange
ipin.05.reset	TEXT	NOPRESET	StringRange
ipin.06.reset	TEXT	NOPRESET	StringRange
ipin.07.reset	TEXT	NOPRESET	StringRange
ipin.08.reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
asiin.01.reset	TEXT	NOPRESET	StringRange
asiin.02.reset	TEXT	NOPRESET	StringRange
asiin.03.reset	TEXT	NOPRESET	StringRange
asiin.04.reset	TEXT	NOPRESET	StringRange
asiin.05.reset	TEXT	NOPRESET	StringRange
asiin.06.reset	TEXT	NOPRESET	StringRange
asiin.07.reset	TEXT	NOPRESET	StringRange
asiin.08.reset	TEXT	NOPRESET	StringRange
asiin.01.enable	CHOICE		EnumRange (ON OFF)
asiin.02.enable	CHOICE		EnumRange (ON OFF)
asiin.03.enable	CHOICE		EnumRange (ON OFF)
asiin.04.enable	CHOICE		EnumRange (ON OFF)
asiin.05.enable	CHOICE		EnumRange (ON OFF)
asiin.06.enable	CHOICE		EnumRange (ON OFF)
asiin.07.enable	CHOICE		EnumRange (ON OFF)
asiin.08.enable	CHOICE		EnumRange (ON OFF)
asiin.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.05.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.06.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.07.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
asiin.08.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.01.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)

name	*type*	*flags*	*range*
ipin.02.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.03.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.04.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.05.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.06.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.07.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.08.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipin.01.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.02.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.03.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.04.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.05.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.06.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.07.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.08.mode	CHOICE		EnumRange (UNICAST MULTICAST)
ipin.01.address	TEXT		StringRange
ipin.02.address	TEXT		StringRange
ipin.03.address	TEXT		StringRange

name	*type*	*flags*	*range*
ipin.04.address	TEXT		StringRange
ipin.05.address	TEXT		StringRange
ipin.06.address	TEXT		StringRange
ipin.07.address	TEXT		StringRange
ipin.08.address	TEXT		StringRange
ipin.01.port	INTEGER		IntegerRange (0 .. 65535)
ipin.02.port	INTEGER		IntegerRange (0 .. 65535)
ipin.03.port	INTEGER		IntegerRange (0 .. 65535)
ipin.04.port	INTEGER		IntegerRange (0 .. 65535)
ipin.05.port	INTEGER		IntegerRange (0 .. 65535)
ipin.06.port	INTEGER		IntegerRange (0 .. 65535)
ipin.07.port	INTEGER		IntegerRange (0 .. 65535)
ipin.08.port	INTEGER		IntegerRange (0 .. 65535)
ipin.01.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.02.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.03.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.04.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.05.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.06.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.07.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.08.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.01.enable	CHOICE		EnumRange (OFF ON)
ipin.02.enable	CHOICE		EnumRange (OFF ON)
ipin.03.enable	CHOICE		EnumRange (OFF ON)
ipin.04.enable	CHOICE		EnumRange (OFF ON)
ipin.05.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
ipin.06.enable	CHOICE		EnumRange (OFF ON)
ipin.07.enable	CHOICE		EnumRange (OFF ON)
ipin.08.enable	CHOICE		EnumRange (OFF ON)
ipin.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.05.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.06.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.07.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.08.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.01.state	TEXT	R/O	StringRange R/O
ipin.02.state	TEXT	R/O	StringRange R/O
ipin.03.state	TEXT	R/O	StringRange R/O
ipin.04.state	TEXT	R/O	StringRange R/O
ipin.05.state	TEXT	R/O	StringRange R/O
ipin.06.state	TEXT	R/O	StringRange R/O
ipin.07.state	TEXT	R/O	StringRange R/O
ipin.08.state	TEXT	R/O	StringRange R/O
ipin.01.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.05.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.06.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.07.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.08.framesLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.05.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.06.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.07.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.08.framesDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.05.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.06.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.07.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.08.framesCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.03.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.05.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.06.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.07.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.08.rtpSequenceErrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.02.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.03.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.04.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.05.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.06.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.07.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.08.maxBurstLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.01.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.02.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.03.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.04.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.05.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.06.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.07.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.08.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipout.01.address	TEXT		StringRange
ipout.02.address	TEXT		StringRange
ipout.03.address	TEXT		StringRange
ipout.04.address	TEXT		StringRange
ipout.05.address	TEXT		StringRange
ipout.06.address	TEXT		StringRange
ipout.07.address	TEXT		StringRange
ipout.08.address	TEXT		StringRange
ipout.01.port	INTEGER		IntegerRange (0 .. 65535)
ipout.02.port	INTEGER		IntegerRange (0 .. 65535)
ipout.03.port	INTEGER		IntegerRange (0 .. 65535)
ipout.04.port	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
ipout.05.port	INTEGER		IntegerRange (0 .. 65535)
ipout.06.port	INTEGER		IntegerRange (0 .. 65535)
ipout.07.port	INTEGER		IntegerRange (0 .. 65535)
ipout.08.port	INTEGER		IntegerRange (0 .. 65535)
ipout.01.enable	CHOICE		EnumRange (OFF ON)
ipout.02.enable	CHOICE		EnumRange (OFF ON)
ipout.03.enable	CHOICE		EnumRange (OFF ON)
ipout.04.enable	CHOICE		EnumRange (OFF ON)
ipout.05.enable	CHOICE		EnumRange (OFF ON)
ipout.06.enable	CHOICE		EnumRange (OFF ON)
ipout.07.enable	CHOICE		EnumRange (OFF ON)
ipout.08.enable	CHOICE		EnumRange (OFF ON)
ipout.01.protocol	CHOICE		EnumRange (RTP UDP)
ipout.02.protocol	CHOICE		EnumRange (RTP UDP)
ipout.03.protocol	CHOICE		EnumRange (RTP UDP)
ipout.04.protocol	CHOICE		EnumRange (RTP UDP)
ipout.05.protocol	CHOICE		EnumRange (RTP UDP)
ipout.06.protocol	CHOICE		EnumRange (RTP UDP)
ipout.07.protocol	CHOICE		EnumRange (RTP UDP)
ipout.08.protocol	CHOICE		EnumRange (RTP UDP)
ipout.01.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.02.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.03.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)

name	*type*	*flags*	*range*
ipout.04.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.05.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.06.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.07.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.08.interface	CHOICE		EnumRange (CONTROL DATA1 DATA2)
ipout.01.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.02.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.03.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.04.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.05.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.06.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.07.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.08.fecMode	CHOICE		EnumRange (OFF COLUMN ONLY COLUMN AND ROW)
ipout.01.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.02.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.03.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.04.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.05.fecColumns	INTEGER		IntegerRange (0 .. 20)

name	*type*	*flags*	*range*
ipout.06.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.07.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.08.fecColumns	INTEGER		IntegerRange (0 .. 20)
ipout.01.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.02.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.03.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.04.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.05.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.06.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.07.fecRows	INTEGER		IntegerRange (0 .. 20)
ipout.08.fecRows	INTEGER		IntegerRange (0 .. 20)
input.asi.01.enable	CHOICE		EnumRange (OFF ON)
input.asi.02.enable	CHOICE		EnumRange (OFF ON)
input.asi.03.enable	CHOICE		EnumRange (OFF ON)
input.asi.04.enable	CHOICE		EnumRange (OFF ON)
input.asi.05.enable	CHOICE		EnumRange (OFF ON)
input.asi.06.enable	CHOICE		EnumRange (OFF ON)
input.asi.07.enable	CHOICE		EnumRange (OFF ON)
input.asi.08.enable	CHOICE		EnumRange (OFF ON)
input.ip.01.enable	CHOICE		EnumRange (OFF ON)
input.ip.02.enable	CHOICE		EnumRange (OFF ON)
input.ip.03.enable	CHOICE		EnumRange (OFF ON)
input.ip.04.enable	CHOICE		EnumRange (OFF ON)
input.ip.05.enable	CHOICE		EnumRange (OFF ON)
input.ip.06.enable	CHOICE		EnumRange (OFF ON)
input.ip.07.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
input.ip.08.enable	CHOICE		EnumRange (OFF ON)
input.asi.01.name	TEXT	R/O	StringRange R/O
input.asi.02.name	TEXT	R/O	StringRange R/O
input.asi.03.name	TEXT	R/O	StringRange R/O
input.asi.04.name	TEXT	R/O	StringRange R/O
input.asi.05.name	TEXT	R/O	StringRange R/O
input.asi.06.name	TEXT	R/O	StringRange R/O
input.asi.07.name	TEXT	R/O	StringRange R/O
input.asi.08.name	TEXT	R/O	StringRange R/O
input.ip.01.name	TEXT	R/O	StringRange R/O
input.ip.02.name	TEXT	R/O	StringRange R/O
input.ip.03.name	TEXT	R/O	StringRange R/O
input.ip.04.name	TEXT	R/O	StringRange R/O
input.ip.05.name	TEXT	R/O	StringRange R/O
input.ip.06.name	TEXT	R/O	StringRange R/O
input.ip.07.name	TEXT	R/O	StringRange R/O
input.ip.08.name	TEXT	R/O	StringRange R/O
input.asi.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.05.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.06.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.07.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.08.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.01.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.02.lock	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
input.ip.03.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.04.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.05.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.06.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.07.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.ip.08.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
input.asi.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.05.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.06.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.07.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.asi.08.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.01.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.02.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.03.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.04.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.05.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.06.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.07.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input.ip.08.totalBitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Stream 1 critical)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Stream 2 critical)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Stream 3 critical)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Stream 4 critical)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Stream 5 critical)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Stream 6 critical)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Stream 7 critical)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Stream 8 critical)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Stream 1 major)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Stream 2 major)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Stream 3 major)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Stream 4 major)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Stream 5 major)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Stream 6 major)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Stream 7 major)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Stream 8 major)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Stream 1 minor)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Stream 2 minor)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Stream 3 minor)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Stream 4 minor)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Stream 5 minor)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Stream 6 minor)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Stream 7 minor)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Stream 8 minor)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Stream 1 warning)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Stream 2 warning)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Stream 3 warning)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Stream 4 warning)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Stream 5 warning)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Stream 6 warning)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Stream 7 warning)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Stream 8 warning)
faults.40	ALARM	R/O	AlarmFlagRange R/O (SNTP sync)
faults.41	ALARM	R/O	AlarmFlagRange R/O (High temperature)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Fan)

name	*type*	*flags*	*range*
faults.43	ALARM	R/O	AlarmFlagRange R/O (Time reference)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Board configuration)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.46	ALARM	R/O	AlarmFlagRange R/O (GPS reference)
faults.47	ALARM	R/O	AlarmFlagRange R/O (GPS sync)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Data 1 interface)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Data 2 interface)

9.445 Newtec-AZ110

This page describes the device driver and the device window for the Newtec Azimuth AZ110 DVB Modulator. The driver accesses the device via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings.
- --- The TX interface settings.
- --- The DVB-S2 specific parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the AZ110.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- amplitudeEQ --- You can set the modulator's amplitude slope equalizer with this parameter (-7 .. +7)
- gainControlMode --- Configures the Gain Control Mode to MGC or AGC
- outputLevelPlan --- Configures the Gain Control Strategy to CONSTANT PWR or CONSTANT RIM.
- levelOffset --- Permits to tweak the output level display by applying an offset. (dB)
- minOutputLevel --- Specifies the minimum value of the output level which can be set with the software. Please note, that this value must be adapted to each individual AZ110 unit to make the driver work properly.
- maxOutputLevel --- Specifies the maximum value of the output level which can be set with the software. Please note, that this value must be adapted to each individual AZ110 unit to make the driver work properly.
- monitorOutput --- Select if the monitor output should also on or follows the tx signal
- ratePriority --- Defines if the symbol rate sets the interface data rate or vice versa.
- extLoFreq --- Lets you apply a frequency offset to the tx frequency parameter.
- extLoPolarity --- Set this to NRM if the RF frequency is above the LO, set it to INV if the RF frequency is below the LO.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.amplitudeEQ	INTEGER	SETUP	IntegerRange (-7 .. 7)
config.gainControlMode	CHOICE	SETUP	EnumRange (MGC AGC)
config.outputLevelPlan	CHOICE	SETUP	EnumRange (CONSTANT PWR CONSTANT RIM)
config.levelOffset	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.minOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.maxOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.hasLevel2	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.monitorOutput	CHOICE	SETUP	EnumRange (ALWAYS ON FOLLOW TX)
config.ratePriority	CHOICE	SETUP	EnumRange (SYMBOL INTERFACE)
config.extLoFreq	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.extLoPolarity	CHOICE	SETUP	EnumRange (NRM INV)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.readBackDelay	INTEGER		IntegerRange (0 .. 0)
info.hwID	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.swID	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.bucketVersion	TEXT	R/O	StringRange R/O
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.occupiedBW	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
modemType	CHOICE		EnumRange (NTC SKYPLEX-SCPC DVB-S DVB-S2)
refClkSrc	CHOICE		EnumRange (INTERN EXTERN)
tx.mod.dataRate	FLOAT	NOPRESET	DoubleRange (0.00000 .. 0.00000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (QPSK-NONE QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10)
tx.mod.rollOff	CHOICE		EnumRange (35% 25% 20%)
tx.mod.clockSource	CHOICE		EnumRange DIS (INTERN EXTERN)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 EXT 204 EXT INTERN)
tx.ifc.hardware	CHOICE		EnumRange (SPI-LVDS RS422-LVDS ASI-A ASI-B OFF)
tx.ifc.asiSwitch	CHOICE		EnumRange (AUTO MANUAL)
tx.ifc.basebandProc	CHOICE		EnumRange (OFF MPEG-RA STUFFING)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.0 .. 5.0)
tx.power.2	FLOAT		DoubleRange DIS (-35.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dvbs2.ACMMode	CHOICE		EnumRange DIS (CCM)

name	*type*	*flags*	*range*
tx.mod.dvbs2.DFLMode	CHOICE		EnumRange DIS (AUTO MANUAL)
tx.mod.dvbs2.basebandDFL	INTEGER		IntegerRange DIS (0 .. 8191)
tx.mod.dvbs2.syncDistance	INTEGER		IntegerRange DIS (0 .. 8191)
tx.mod.dvbs2.isi	TEXT		StringRange DIS
tx.mod.dvbs2.insertPilot	CHOICE		EnumRange DIS (ON OFF)
tx.mod.dvbs2.FECFrameType	CHOICE		EnumRange DIS (SHORT NORMAL)
tx.mod.dvbs2.physEfficiency	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
tx.mod.dvbs2.baseEfficiency	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
tx.mod.dvbs2.shortFrames	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Input framing)
faults.10	ALARM	R/O	AlarmFlagRange R/O (ASI code violations)
faults.11	ALARM	R/O	AlarmFlagRange R/O (ASI opt. sig. det.)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LVDS signal detect)
faults.13	ALARM	R/O	AlarmFlagRange R/O (NCR inserter GPS 1pps)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Baseband frame sync)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Buffer underflow)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Buffer overflow)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Clock PLL)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Synthesiser)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF phase lock DRO)
faults.20	ALARM	R/O	AlarmFlagRange R/O (BISS summary)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Internal MC module)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Interface module)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Internal modulator)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
internal.tx.on	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)

9.446 Newtec-AZ200-1-To-1

The *Newtec-AZ200-1-To-1* device driver controls a single 1:1 redundancy implemented with a Newtec AZ200 switch controller. The driver is targeted to operational use, it does not implement a configuration interface to the AZ200 switch controller. The driver uses SNMP to control the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to reflect the configuration of the redundancy switch in the sat-nms software.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- functionIndex --- The Newtec Azimuth switch controller is capable to comprise up to 5 redundancy switching entities called "functions". With this parameter (1..5) you select the

function to address with this device instance. With a single 1:1 redundancy switch set this to '1'

- chain1Fault --- The Newtec Azimuth switch controller provides 9 fault inputs which may be configured to act as triggers for the redundancy switching. Select the fault input which is used in the AZ200 to report a fault of modem/chain 1 here in order to show this fault condition at the first page of the device window. The display of the fault condition is not affected by the fault classification setting described below in section "Fault classifications".
- chain2Fault --- Like chain1Fault, bot for modem/chain 2.

Fault classification

The Newtec Azimuth switch controller reports faults of the modems it supervises in the same way as internal faults. In most cases this behaviour is not desirable, as the switch controller summary fault should indicate a fault of the switch controller itself, any faults of the modems connected to the controller are reported separately by the software. For this reason, it is recommended to set the fault classification of the faults "User 1" .. "User 9" to "NONE" at the faults page of the device window.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwID	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.swID	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
config.functionIndex	INTEGER	SAVE SETUP	IntegerRange (1 .. 5)

name	*type*	*flags*	*range*
config.chain1Fault	CHOICE	SAVE SETUP	EnumRange (NONE USER 1 USER 2 USER 3 USER 4 USER 5 USER 6 USER 7 USER 8 USER 9)
config.chain2Fault	CHOICE	SAVE SETUP	EnumRange (NONE USER 1 USER 2 USER 3 USER 4 USER 5 USER 6 USER 7 USER 8 USER 9)
location	CHOICE		EnumRange (1 2)
switchMode	CHOICE		EnumRange (MANUAL AUTO)
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Supply voltage)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Redundant PSU)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Red. dev. comm.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (User 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (User 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (User 3)
faults.12	ALARM	R/O	AlarmFlagRange R/O (User 4)
faults.13	ALARM	R/O	AlarmFlagRange R/O (User 5)
faults.14	ALARM	R/O	AlarmFlagRange R/O (User 6)
faults.15	ALARM	R/O	AlarmFlagRange R/O (User 7)
faults.16	ALARM	R/O	AlarmFlagRange R/O (User 8)
faults.17	ALARM	R/O	AlarmFlagRange R/O (User 9)
faults.18	ALARM	R/O	AlarmFlagRange R/O (M&C module)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Extension units)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Architecture)

9.447 Newtec-AZ410

This page describes the device driver and the device window for the Newtec Azimuth AZ410 DVB Modem. The driver accesses the device via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings.
- --- The DVB-S2 specific parameters
- --- The demodulator settings.
- --- The TX interface settings.
- --- The network routes.
- --- The meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the AZ410.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- rx.1.lnbPower --- Switches the LNB power for input 1 ON or OFF
- rx.1.lnbVoltage --- Selects the LNB voltage and 22kHz tone for input 1
- rx.1.lnbTone -- Switches the LNB tone (22kHz) for input 1 ON or OFF
- rx.1.spectrInv --- Selects if to invert the lo frequency calculation for input 1. Select NORM if the signal frequency is above the LO, INVERT if the signal frequency is below the LO.
- rx.2.lnbPower --- Switches the LNB power for input 2 ON or OFF
- rx.2.lnbVoltage --- Selects the LNB voltage and 22kHz tone for input 2
- rx.2.lnbTone -- Switches the LNB tone (22kHz) for input 2 ON or OFF
- rx.2.spectrInv --- Selects if to invert the lo frequency calculation for input 2. Select NORM if the signal frequency is above the LO, INVERT if the signal frequency is below the LO.
- rx.loFreqHigh --- Defines the high band receive lo frequency, applicable for both inputs.
- rx.loFreqLow --- Defines the low band receive lo frequency, applicable for both inputs.
- rx.bandEdge --- Defines the band edge frequency.
- tx.loFreq --- Defines the transmit (BUC) lo frequency.
- tx.loPolarity --- Selects if to invert the transmit lo frequency calculation. Select NORM if the signal frequency is above the LO, INVERT if the signal frequency is below the LO.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.3V3.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.5V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.12V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.m12V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.rx.1.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.1.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.rx.1.lnbTone	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.1.spectrInv	CHOICE	SETUP	EnumRange (NORM INVERT)
config.rx.2.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.2.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V/22KHz)
config.rx.2.lnbTone	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.2.spectrInv	CHOICE	SETUP	EnumRange (NORM INVERT)
config.rx.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rx.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rx.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.tx.loFreq	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.tx.loPolarity	CHOICE	SAVE SETUP	EnumRange (NORM INVERT)
internal.dummy	TEXT	R/O	StringRange R/O
internal.lo	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.freq	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.dvbStreams	TEXT	R/O	StringRange R/O
modemType	CHOICE		EnumRange (IP MPE: eth(ip)-air(ts) IP XPE/GSE: eth(ip)-air(s2bbf) DUALFLOW: eth(ip)/asi(ts)-air(s2bbf/ts) TSoIP: eth(tsoip)-asi(ts) ASI: asi(ts)-air(ts) ASI FIXED DELAY: asi(ts)-air(ts) ASI MULTISTREAM: asi(ts)-air(ts))
dvbMode	CHOICE		EnumRange (DVB-S DVB-S2)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.6 .. 5.4)
tx.gainControl	CHOICE		EnumRange (MGC ACG)
tx.outputLevelPlan	CHOICE		EnumRange (CONST PWR CONST RIM)
tx.mod.acmMode	CHOICE		EnumRange (CCM VCM ACM)
tx.mod.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)

name	*type*	*flags*	*range*
tx.mod.insertPilots	CHOICE		EnumRange (ON OFF)
tx.mod.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.dataRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.mod.symbolRate	FLOAT	NOPRESET	DoubleRange (0.00000 .. 0.00000)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.isi	TEXT		StringRange
tx.mod.plsSignature	INTEGER		IntegerRange (0 .. 262141)
tx.mod.occupiedBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORMAL INVERT)
tx.mod.amplitudeEQ	INTEGER		IntegerRange (-7 .. 7)
tx.mod.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
tx.dvbs2str.1.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.1.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.1.isi	TEXT		StringRange

name	*type*	*flags*	*range*
tx.dvbs2str.1.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.1.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.2.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.2.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.2.isi	TEXT		StringRange
tx.dvbs2str.2.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.2.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.3.enable	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
tx.dvbs2str.3.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.3.isi	TEXT		StringRange
tx.dvbs2str.3.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.3.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.4.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.4.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.4.isi	TEXT		StringRange
tx.dvbs2str.4.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.4.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)

name	*type*	*flags*	*range*
tx.dvbs2str.5.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.5.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.5.isi	TEXT		StringRange
tx.dvbs2str.5.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.5.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.6.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.6.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.6.isi	TEXT		StringRange
tx.dvbs2str.6.insertPilots	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
tx.dvbs2str.6.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.7.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.7.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.7.isi	TEXT		StringRange
tx.dvbs2str.7.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.7.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.8.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.8.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.8.isi	TEXT		StringRange

name	*type*	*flags*	*range*
tx.dvbs2str.8.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.8.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.9.enable	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.9.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.9.isi	TEXT		StringRange
tx.dvbs2str.9.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.9.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.dvbs2str.10.enable	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
tx.dvbs2str.10.type	CHOICE		EnumRange (QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.dvbs2str.10.isi	TEXT		StringRange
tx.dvbs2str.10.insertPilots	CHOICE		EnumRange (ON OFF)
tx.dvbs2str.10.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
rx.input	CHOICE		EnumRange (IFL1 IFL2)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.dem.acmMode	CHOICE		EnumRange (CCM AUTO-CCM ACM)
rx.dem.dataRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.dem.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.dem.acqRange	INTEGER		IntegerRange (1 .. 5000)
rx.dem.isi	TEXT		StringRange
rx.dem.isiControl	CHOICE		EnumRange (OFF ON)
rx.dem.pidFiltering	CHOICE		EnumRange (OFF ON)
rx.dem.spectrumInvert	CHOICE		EnumRange (AUTO NORMAL INVERT)
rx.dem.plsSignature	INTEGER		IntegerRange (0 .. 262141)

name	*type*	*flags*	*range*
rx.dem.rollOff	CHOICE		EnumRange (AUTO 5% 10% 15% 20% 25% 35%)
rx.dem.type	CHOICE		EnumRange (DUMMY-PL QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-AUTO QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 16QAM-AUTO 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-AUTO 8PSK-5/6 8PSK-8/9 8PSK-9/10 AUTO)
rx.dem.pilots	CHOICE		EnumRange (OFF ON)
rx.dem.fecFrame	CHOICE		EnumRange (NORMAL SHORT)
rx.backup.1.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.backup.1.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.backup.2.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.backup.2.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.backup.switchDelay	INTEGER		IntegerRange (10 .. 120)
rx.backup.switchEnable	CHOICE		EnumRange (OFF ON)
rx.backup.switchCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.mtu	INTEGER		IntegerRange (1500 .. 2000)

name	*type*	*flags*	*range*
eth.enableEthItf	CHOICE		EnumRange (NONE ITF-A ITF-B)
eth.encapProtocol	CHOICE		EnumRange (NONE ULE DATAPIPING XPE MPE GSE)
eth.bridgeType	CHOICE		EnumRange (L2 ETH-BRIDGE L3 IP-BRIDGE L3 IP-ROUTER)
eth.vlanSupport	CHOICE		EnumRange (OFF ON)
eth.mgmntEncapStreamId	INTEGER		IntegerRange (1 .. 35)
eth.l2Routing	CHOICE		EnumRange (OFF ON)
eth.promiscuousMode	CHOICE	R/O	EnumRange R/O (OFF ON)
eth.forcePromiscuous	CHOICE		EnumRange (AUTOMATIC FORCE)
eth.rxr.1.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.1.destIp	TEXT		StringRange
eth.rxr.1.ipNetMask	TEXT		StringRange
eth.rxr.1.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.1.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.2.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.2.destIp	TEXT		StringRange
eth.rxr.2.ipNetMask	TEXT		StringRange
eth.rxr.2.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.2.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.3.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.3.destIp	TEXT		StringRange
eth.rxr.3.ipNetMask	TEXT		StringRange
eth.rxr.3.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.3.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.4.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
eth.rxr.4.destIp	TEXT		StringRange
eth.rxr.4.ipNetMask	TEXT		StringRange
eth.rxr.4.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.4.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.5.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.5.destIp	TEXT		StringRange
eth.rxr.5.ipNetMask	TEXT		StringRange
eth.rxr.5.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.5.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.6.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.6.destIp	TEXT		StringRange
eth.rxr.6.ipNetMask	TEXT		StringRange
eth.rxr.6.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.6.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.7.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.7.destIp	TEXT		StringRange
eth.rxr.7.ipNetMask	TEXT		StringRange
eth.rxr.7.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.7.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.8.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.8.destIp	TEXT		StringRange
eth.rxr.8.ipNetMask	TEXT		StringRange
eth.rxr.8.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.8.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.9.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.9.destIp	TEXT		StringRange

name	*type*	*flags*	*range*
eth.rxr.9.ipNetMask	TEXT		StringRange
eth.rxr.9.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.9.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.rxr.10.enable	CHOICE		EnumRange (OFF ON)
eth.rxr.10.destIp	TEXT		StringRange
eth.rxr.10.ipNetMask	TEXT		StringRange
eth.rxr.10.streamId	INTEGER		IntegerRange (1 .. 35)
eth.rxr.10.packingDelay	INTEGER		IntegerRange (0 .. 1000)
eth.dmr.1.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.1.isi	TEXT		StringRange
eth.dmr.2.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.2.isi	TEXT		StringRange
eth.dmr.3.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.3.isi	TEXT		StringRange
eth.dmr.4.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.4.isi	TEXT		StringRange
eth.dmr.5.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.5.isi	TEXT		StringRange
eth.dmr.6.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.6.isi	TEXT		StringRange
eth.dmr.7.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.7.isi	TEXT		StringRange
eth.dmr.8.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.8.isi	TEXT		StringRange
eth.dmr.9.enable	CHOICE		EnumRange (OFF ON)
eth.dmr.9.isi	TEXT		StringRange
eth.dmr.10.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
eth.dmr.10.isi	TEXT		StringRange
asi.itfRedundancy	CHOICE		EnumRange (OFF ON ON PRIO-A ON PRIO-B)
asi.1.in.dvbs2StreamId	INTEGER		IntegerRange (1 .. 35)
asi.1.in.dvbs2PackingDelay	INTEGER		IntegerRange (0 .. 0)
asi.1.in.mpegFraming	CHOICE		EnumRange (188 204 UNFRAMED)
asi.1.in.dataRate	FLOAT		DoubleRange (0.05500 .. 120.00000)
asi.1.in.limitRate	CHOICE		EnumRange (OFF ON)
asi.1.in.dvbs2NpdMode	CHOICE		EnumRange (OFF ON)
asi.1.in.dvbs2MaxNpd	INTEGER		IntegerRange (0 .. 255)
asi.1.in.dvbs2IssyMode	CHOICE		EnumRange (OFF SHORT LONG)
asi.1.in.rateAdapter	CHOICE		EnumRange (OFF ON ON NPD)
asi.1.in.signalSelect	CHOICE		EnumRange (OFF ON TESTGEN)
asi.1.out.dvbs2Isi	TEXT		StringRange
asi.1.out.mpegFraming	CHOICE		EnumRange (188 204 INTERNAL)
asi.1.out.dataRate	FLOAT		DoubleRange (0.05500 .. 216.00000)
asi.1.out.timingRecov	CHOICE		EnumRange (OFF ON ISSY)
asi.1.out.signalSelect	CHOICE		EnumRange (OFF ON TESTGEN)
asi.1.gen.type	CHOICE		EnumRange (PBRS COUNTER)
asi.1.gen.pidInsert	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
asi.1.gen.pid	INTEGER		IntegerRange (0 .. 65535)
asi.1.gen.dPackets	INTEGER		IntegerRange (1 .. 255)
asi.1.gen.nPackets	INTEGER		IntegerRange (0 .. 255)
asi.1.gen.outputRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asi.2.in.dvbs2StreamId	INTEGER		IntegerRange (1 .. 35)
asi.2.in.dvbs2PackingDelay	INTEGER		IntegerRange (0 .. 0)
asi.2.in.mpegFraming	CHOICE		EnumRange (188 204 UNFRAMED)
asi.2.in.dataRate	FLOAT		DoubleRange (0.05500 .. 120.00000)
asi.2.in.limitRate	CHOICE		EnumRange (OFF ON)
asi.2.in.dvbs2NpdMode	CHOICE		EnumRange (OFF ON)
asi.2.in.dvbs2MaxNpd	INTEGER		IntegerRange (0 .. 255)
asi.2.in.dvbs2IssyMode	CHOICE		EnumRange (OFF SHORT LONG)
asi.2.in.rateAdapter	CHOICE		EnumRange (OFF ON ON NPD)
asi.2.in.signalSelect	CHOICE		EnumRange (OFF ON TESTGEN)
asi.2.out.dvbs2Isi	TEXT		StringRange
asi.2.out.mpegFraming	CHOICE		EnumRange (188 204 INTERNAL)
asi.2.out.dataRate	FLOAT		DoubleRange (0.05500 .. 216.00000)
asi.2.out.timingRecov	CHOICE		EnumRange (OFF ON ISSY)
asi.2.out.signalSelect	CHOICE		EnumRange (OFF ON TESTGEN)
asi.2.gen.type	CHOICE		EnumRange (PBRS COUNTER)
asi.2.gen.pidInsert	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
asi.2.gen.pid	INTEGER		IntegerRange (0 .. 65535)
asi.2.gen.dPackets	INTEGER		IntegerRange (1 .. 255)
asi.2.gen.nPackets	INTEGER		IntegerRange (0 .. 255)
asi.2.gen.outputRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rx.meter.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.type	TEXT	R/O	StringRange R/O
rx.meter.frameType	TEXT	R/O	StringRange R/O
rx.meter.pilots	TEXT	R/O	StringRange R/O
rx.meter.rollOff	TEXT	R/O	StringRange R/O
rx.meter.isiFilterByte	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.bbDataFieldLength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.ifc.pidFilterMode	CHOICE		EnumRange (OFF ON)
rx.ifc.pidFilterCode	INTEGER		IntegerRange (0 .. 8191)
rx.ifc.pbrsStatus	TEXT	R/O	StringRange R/O
rx.ifc.errorCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.ifc.pbrsSyncSeconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.ifc.errorRate	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)
rx.ifc.errorRatio	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)

name	*type*	*flags*	*range*
rx.ifc.pbrsMonitorInput	CHOICE		EnumRange (OFF ASI1 RX ASI1 TX ASI2 RX ASI2 TX ASI3 RX ASI3 TX)
ethIfcAState	TEXT	R/O	StringRange R/O
ethIfcBState	TEXT	R/O	StringRange R/O
eth.errorPackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.packets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.resetCounters	CHOICE	NOCOMPARE	EnumRange (PERIOD NEVER ONCE)
rx.bb.txTsFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxTsFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxDroppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.tsCrc8Errors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.bbfCrc8Errors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.resetCounters	CHOICE	NOCOMPARE	EnumRange (PERIOD NEVER ONCE)
rx.bb.rxRawBitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxRawFramerate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.1.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.2.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.3.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General device)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Eth IfA Link)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Eth IfB Link)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Eth IfA half duplex)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Eth IfB half duplex)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IP GW unreachable)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Eth Ifc switchover)
faults.15	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 1 Input)
faults.16	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 2 Input)
faults.17	ALARM	R/O	AlarmFlagRange R/O (ASI Ifc switchover)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Demod Rx bad packet)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Demod Rx buf oflow)
faults.20	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 1 buf uflow)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 2 buf uflow)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 3 buf uflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 4 buf uflow)
faults.24	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 1 buf oflow)
faults.25	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 2 buf oflow)
faults.26	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 3 buf oflow)
faults.27	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 4 buf oflow)
faults.28	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 1 buf oflow)
faults.29	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 2 buf oflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (BB frame sync)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Clock PLL)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Demod lock)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Demod tuner lock)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Demod PL sync)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Demod BB sync)

name	*type*	*flags*	*range*
faults.37	ALARM	R/O	AlarmFlagRange R/O (Internal M&C module)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Interface module)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Internal modulator)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Demodulator module)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Internal communication)
faults.43	ALARM	R/O	AlarmFlagRange R/O (NCR sync status)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Input alarm)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Level saturation)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Mod bufer oflow)

9.448 Newtec-AZ910

This page describes the device driver and the device window for the Newtec Azimuth AZ910 DVB-S2 Demodulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency etc.) and some global settings.
- --- The backup carrier configuration.
- --- The demodulator settings.
- --- The interface settings (eth, asi)
- --- The receive status parameters including some interface statistics
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Newtec EL470.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- rx.1.lnbPower --- Switches the LNB power for input 1 ON or OFF.
- rx.1.lnbVoltge --- Selects voltage and 22kHz tone for input 1.
- rx.1.spectrInv --- The LO spectrum invert for input 1. Set to NORM if the LO frequency is below the receive frequency, set to INVERT if the LO is above.
- rx.2.lnbPower --- Switches the LNB power for input 2 ON or OFF.
- rx.2.lnbVoltge --- Selects voltage and 22kHz tone for input 2.
- rx.2.spectrInv --- The LO spectrum invert for input 2. Set to NORM if the LO frequency is below the receive frequency, set to INVERT if the LO is above.
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.3V3.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.5V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.12V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.m12V.supply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.rx.1.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.1.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V/0KHz 13V/22KHz 18V/0KHz 18V/22KHz)
config.rx.1.spectrInv	CHOICE	SETUP	EnumRange (NORM INVERT)
config.rx.2.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.2.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V/0KHz 13V/22KHz 18V/0KHz 18V/22KHz)
config.rx.2.spectrInv	CHOICE	SETUP	EnumRange (NORM INVERT)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
internal.dummy	TEXT	R/O	StringRange R/O
internal.lo	FLOAT	R/O	DoubleRange R/O (0 .. 0)
modemType	CHOICE		EnumRange (IP MPE: eth(ip)-air(ts) IP XPE/GSE: eth(ip)-air(s2bbf) DUALFLOW: eth(ip)/asi(ts)-air(s2bbf/ts) TSoIP: eth(tsoip)-asi(ts) ASI: asi(ts)-air(ts) ASI FIXED DELAY: asi(ts)-air(ts) ASI MULTISTREAM: asi(ts)-air(ts))
dvbMode	CHOICE		EnumRange (DVB-S DVB-S2)
rx.input	CHOICE		EnumRange (IFL1 IFL2)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.dem.acmMode	CHOICE		EnumRange (CCM AUTO-CCM ACM)
rx.dem.dataRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.dem.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.dem.acqRange	INTEGER		IntegerRange (1 .. 5000)
rx.dem.isiControl	CHOICE		EnumRange (OFF ON)
rx.dem.pidFiltering	CHOICE		EnumRange (OFF ON)
rx.dem.plsSignature	INTEGER		IntegerRange (0 .. 262141)
rx.dem.rollOff	CHOICE		EnumRange (AUTO 5% 10% 15% 20% 25% 35%)

name	*type*	*flags*	*range*
rx.dem.type	CHOICE		EnumRange (DUMMY-PL QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-AUTO QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 16QAM-AUTO 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-AUTO 8PSK-5/6 8PSK-8/9 8PSK-9/10 AUTO)
rx.dem.pilots	CHOICE		EnumRange (OFF ON)
rx.dem.fecFrame	CHOICE		EnumRange (NORMAL SHORT)
rx.backup.1.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.backup.1.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.backup.2.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.backup.2.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.backup.switchDelay	INTEGER		IntegerRange (10 .. 120)
rx.backup.switchEnable	CHOICE		EnumRange (OFF ON)
rx.backup.switchCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.mtu	INTEGER		IntegerRange (1500 .. 2000)
eth.l2Routing	CHOICE		EnumRange (OFF ON)
eth.promiscuousMode	CHOICE	R/O	EnumRange R/O (OFF ON)
eth.forcePromiscuous	CHOICE		EnumRange (AUTOMATIC FORCE)

name	*type*	*flags*	*range*
asi.1.out.dvbs2Isi	TEXT		StringRange
asi.1.out.mpegFraming	CHOICE		EnumRange (188 204 INTERNAL)
asi.1.out.dataRate	FLOAT		DoubleRange (0.05500 .. 216.00000)
asi.1.out.timingRecov	CHOICE		EnumRange (OFF ON ISSY)
asi.1.out.signalSelect	CHOICE		EnumRange (OFF RXSIGNAL TESTGEN)
asi.1.gen.type	CHOICE		EnumRange (PBRS COUNTER)
asi.1.gen.pidInsert	CHOICE		EnumRange (OFF ON)
asi.1.gen.pid	INTEGER		IntegerRange (0 .. 65535)
asi.1.gen.dPackets	INTEGER		IntegerRange (1 .. 255)
asi.1.gen.nPackets	INTEGER		IntegerRange (0 .. 255)
asi.1.gen.outputRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
asi.2.out.dvbs2Isi	TEXT		StringRange
asi.2.out.mpegFraming	CHOICE		EnumRange (188 204 INTERNAL)
asi.2.out.dataRate	FLOAT		DoubleRange (0.05500 .. 216.00000)
asi.2.out.timingRecov	CHOICE		EnumRange (OFF ON ISSY)
asi.2.out.signalSelect	CHOICE		EnumRange (OFF RXSIGNAL TESTGEN)
asi.2.gen.type	CHOICE		EnumRange (PBRS COUNTER)
asi.2.gen.pidInsert	CHOICE		EnumRange (OFF ON)
asi.2.gen.pid	INTEGER		IntegerRange (0 .. 65535)
asi.2.gen.dPackets	INTEGER		IntegerRange (1 .. 255)

name	*type*	*flags*	*range*
asi.2.gen.nPackets	INTEGER		IntegerRange (0 .. 255)
asi.2.gen.outputRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rx.meter.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.type	TEXT	R/O	StringRange R/O
rx.meter.frameType	TEXT	R/O	StringRange R/O
rx.meter.pilots	TEXT	R/O	StringRange R/O
rx.meter.rollOff	TEXT	R/O	StringRange R/O
rx.meter.isiFilterByte	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.bbDataFieldLength	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.ifc.pidFilterMode	CHOICE		EnumRange (OFF ON)
rx.ifc.pidFilterCode	INTEGER		IntegerRange (0 .. 8191)
rx.ifc.pbrsStatus	TEXT	R/O	StringRange R/O
rx.ifc.errorCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.ifc.pbrsSyncSeconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.ifc.errorRate	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)
rx.ifc.errorRatio	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)

name	*type*	*flags*	*range*
rx.ifc.pbrsMonitorInput	CHOICE		EnumRange (OFF ASI1 RX ASI1 TX ASI2 RX ASI2 TX ASI3 RX ASI3 TX)
ethIfcAState	TEXT	R/O	StringRange R/O
ethIfcBState	TEXT	R/O	StringRange R/O
eth.errorPackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.packets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
eth.resetCounters	CHOICE	NOCOMPARE	EnumRange (PERIOD NEVER ONCE)
rx.bb.txTsFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxTsFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxDroppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.tsCrc8Errors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.bbfCrc8Errors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.resetCounters	CHOICE	NOCOMPARE	EnumRange (PERIOD NEVER ONCE)
rx.bb.rxRawBitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.rxRawFramerate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.1.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.2.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.bb.3.droppedFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General device)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Eth IfA Link)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Eth IfB Link)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Eth IfA half duplex)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Eth IfB half duplex)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IP GW unreachable)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Eth Ifc switchover)
faults.15	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 1 Input)
faults.16	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 2 Input)
faults.17	ALARM	R/O	AlarmFlagRange R/O (ASI Ifc switchover)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Demod Rx bad packet)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Demod Rx buf oflow)
faults.20	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 1 buf uflow)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 2 buf uflow)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 3 buf uflow)
faults.23	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 4 buf uflow)
faults.24	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 1 buf oflow)
faults.25	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 2 buf oflow)
faults.26	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 3 buf oflow)
faults.27	ALARM	R/O	AlarmFlagRange R/O (ASI Tx 4 buf oflow)
faults.28	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 1 buf oflow)
faults.29	ALARM	R/O	AlarmFlagRange R/O (ASI Rx 2 buf oflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Demod lock)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Demod tuner lock)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Demod PL sync)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Demod BB sync)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Internal M&C module)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Interface module)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Demodulator module)

name	*type*	*flags*	*range*
faults.37	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Internal communication)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Level saturation)

9.449 Newtec-EL470

This page describes the device driver and the device window for the Newtec EL470 DVB-S2 Modem. The driver is not a full implementation of all parameters provided by the modem, it only contains a number of parameters required for operational use of the modem. The driver controls the modem via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings.
- --- The demodulator settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Newtec EL470.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read

community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- tx.loFreq --- The LO frequency to be used with the L-band output of the device. Entering the LO frequency of the BUC here (MHz), permits to display/set the transmit frequency in terms of true RF frequency. Setting this parameter to 0 lets the software display the TX frequency setting as the L-band frequency.
- rx.loFreq1 --- The LO frequency to be used with the L-band input 1 of the device. Entering the LO frequency of the LNB here (MHz), permits to display/set the receive frequency in terms of true RF frequency. Setting this parameter to 0 lets the software display the RX frequency setting as the L-band frequency.
- rx.loFreq2 --- Like rx.loFreq1, but for the L-band input 2 of the device.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.tx.loFreq	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.rx.loFreq1	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.rx.loFreq2	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.readBackDelay	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
info.hwID	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.swID	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
dvbMode	CHOICE		EnumRange (DVB-S DVB-S2)
modemType	CHOICE		EnumRange (Eth(IP)<->Air(TS) Eth(IP) <->Air(XPE) Eth(IP)/ASI(TS)<- >Air(XPE/TS) Eth(TSoIP)<->ASI(TS) Eth(TSoIP)<->Air(TS) ASI(TS)<- >Air(TS) ASI(TS)<->Air(TS)(fixed delay) ASI(S2BBF)<->Air(S2BBF) Eth(S2BBF)<->Air(S2BBF) ASI(TS)<- >Air(TS)(multistream) Eth(ntS2BBF)- >Air(S2BBF))
ethIfcAState	TEXT	R/O	StringRange R/O
ethIfcBState	TEXT	R/O	StringRange R/O
rx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
rx.input	CHOICE		EnumRange (IFL1 IFL2 IF)
rx.dem.acmMode	CHOICE		EnumRange (CCM AUTO-CCM ACM)
rx.dem.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
rx.dem.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
rx.dem.rollOff	CHOICE		EnumRange (35% 25% 20%)

name	*type*	*flags*	*range*
rx.dem.modCod	CHOICE		EnumRange DIS (Dummy PLFRAMES QPSK NO FEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-AUTO QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 16QAM-AUTO 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-auto 8PSK-5/6 8PSK-8/9 8PSK-9/10 AUTO)
rx.dem.pilots	CHOICE		EnumRange (OFF ON)
rx.dem.fecFrame	CHOICE		EnumRange (NORMAL SHORT)
rx.dem.acqRange	FLOAT		DoubleRange (1.0 .. 5000.0)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.mod.acmMode	CHOICE		EnumRange (CCM VCM ACM)
tx.mod.rollOff	CHOICE		EnumRange (35% 25% 20%)
tx.mod.occupiedBw	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
tx.mod.minModCod	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 QPSK-13/15 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 16APSK-13/15 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 32APSK-13/15 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 8PSK-13/15)
tx.mod.maxModCod	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 QPSK-13/15 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 16APSK-13/15 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 32APSK-13/15 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 8PSK-13/15)
tx.mod.failModCod	CHOICE		EnumRange (LOWEST QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 QPSK-13/15 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 16APSK-13/15 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 32APSK-13/15 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 8PSK-13/15)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
internal.tx.on	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.rx.acm	TEXT	R/O	StringRange R/O

9.450 Newtec-FRC0710

This page describes the device driver and the device window for the Newtec FRC0710 upconverter via SNMP.

This driver was successfully tested with the AZ710 unit as well.

- FRC0710: SW version v10.0.6 (DBS Band)
- AZ 710: SW version v8.0.4 (DBS Band)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- Transmit parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the FRC0710

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- powerDisplayMode --- Change between gain or attenuation display.

- inputHiThreshold --- The level threshold for the 'Input overload' fault.
- inputLoThreshold --- The level threshold for the 'Input underload' fault.
- inputHiFault --- Configures the device whether to signal an 'Input overload' fault if the input level exceeds the threshold value.
- inputLoFault --- Configures the device whether to signal an 'Input underload' fault if the input level falls below the threshold value (The device only signals this fault if the output is enabled and at least one input is switched on).
- loPolarity --- Change LO polarity between Normal and Inverted (NRM/INV)
- autoSave --- Switching this to YES makes the driver automatically save the device configuration into it's flash memory after every change. NO disables automatic saving, you may use the 'SAVE BOOT CONFIG' button in the sat-nms GUI to save the actual configuration explicitly.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Parameter storage in the FRC0710

Any changes made to the parameters of the FRC0710 only are valid until the device reboots - unless they are stored into the 'boot configuration' flash memory of the device.

The sat-nms driver provides two methods to handle this: In automatic mode the driver commands a 'save boot configuration' after each change made to a parameter. In manual mode, you have to store the parameter by clicking the 'SAVE BOOT CONFIG' button in the sat-nms GUI to save the actual configuration explicitly.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.deviceName	TEXT	R/O	StringRange R/O
info.loFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu.3V3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu.5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu.12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.psu.neg12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.inputHiThresh	FLOAT	SETUP	DoubleRange (0 .. 0)
config.inputLoThresh	FLOAT	SETUP	DoubleRange (0 .. 0)
config.inputHiFault	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.inputLoFault	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.loPolarity	CHOICE	SETUP	EnumRange (NRM INV)
config.autoSave	CHOICE	SETUP	EnumRange (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.int.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
tx.inputGain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.inputAttn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.00 .. 0.00)
tx.lb.frequency	FLOAT		DoubleRange (0.00 .. 0.00)
tx.if.frequency	CHOICE		EnumRange (70 MHz 140 MHz)
tx.inputBw	CHOICE		EnumRange (40 MHz 80 MHz)
meas.if.txLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.int.txLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.txLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
saveBootConfig	TEXT	NOPRESET	StringRange
internal.save	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (IF to L-band Out of lock LO1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (IF to L-band Out of lock LO2)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (IF to L-band Input overload)
faults.12	ALARM	R/O	AlarmFlagRange R/O (IF to L-band Input underload)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Upconverter RF DRO 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Upconverter Amplifier current)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Internal M&C module)
faults.16	ALARM	R/O	AlarmFlagRange R/O (IF To L-band module)
faults.17	ALARM	R/O	AlarmFlagRange R/O (RF module)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Internal communication)

9.451 Newtec-FRC0750

This page describes the device driver and the device window for the Newtec FRC0750 block upconverter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Device settings and status display
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the FRC0750

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- rfHiThreshold --- The level threshold for the 'Output overload' fault.
- rfLoThreshold --- The level threshold for the 'Output underload' fault.
- rfHiFault --- Configures the device whether to signal an 'Output overload' fault if the output level exceeds the threshold value.
- rfLoFault --- Configures the device whether to signal an 'Output underload' fault if the output level falls below the threshold value (The device only signals this fault if the output is enabled and at least one input is switched on).
- autoSave --- Switching this to YES makes the driver automatically save the device configuration into it's flash memory after every change. NO disables automatic saving, you may use the 'SAVE BOOT CONFIG' button in the sat-nms GUI to save the actual configuration explicitly.

Parameter storage in the FRC0750

Any changes made to the parameters of the FRC0750 only are valid until the device reboots - unless they are stored into the 'boot configuration' flash memory of the device.

The sat-nms driver provides two methods to handle this: In automatic mode the driver commands a 'save boot configuration' after each change made to a parameter. In manual mode, you have to store the parameter by clicking the 'SAVE BOOT CONFIG' button in the sat-nms GUI to save the actual configuration explicitly.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.deviceName	TEXT	R/O	StringRange R/O
info.rfFreqRange	TEXT	R/O	StringRange R/O
info.loFrequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.rfHiThresh	FLOAT	SETUP	DoubleRange (0 .. 0)
config.rfLoThresh	FLOAT	SETUP	DoubleRange (0 .. 0)
config.rfHiFault	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.rfLoFault	CHOICE	SETUP	EnumRange (DISABLED ENABLED)
config.inputs	CHOICE	SAVE SETUP	EnumRange (4 8)
config.autoSave	CHOICE	SETUP	EnumRange (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
rfOut.on	CHOICE		EnumRange (OFF ON)
rfOut.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
lbOut.on	CHOICE		EnumRange (OFF ON)
input.1.on	CHOICE		EnumRange (OFF ON)
input.1.gain	FLOAT		DoubleRange (-30.0 .. 10.0)
input.2.on	CHOICE		EnumRange (OFF ON)
input.2.gain	FLOAT		DoubleRange (-30.0 .. 10.0)
input.3.on	CHOICE		EnumRange (OFF ON)
input.3.gain	FLOAT		DoubleRange (-30.0 .. 10.0)
input.4.on	CHOICE		EnumRange (OFF ON)
input.4.gain	FLOAT		DoubleRange (-30.0 .. 10.0)
input.5.on	CHOICE		EnumRange DIS (OFF ON)
input.5.gain	FLOAT		DoubleRange DIS (-30.0 .. 10.0)
input.6.on	CHOICE		EnumRange DIS (OFF ON)
input.6.gain	FLOAT		DoubleRange DIS (-30.0 .. 10.0)
input.7.on	CHOICE		EnumRange DIS (OFF ON)
input.7.gain	FLOAT		DoubleRange DIS (-30.0 .. 10.0)
input.8.on	CHOICE		EnumRange DIS (OFF ON)
input.8.gain	FLOAT		DoubleRange DIS (-30.0 .. 10.0)
state.rfOut	BOOLEAN	R/O	BooleanRange R/O (true false)
state.lbOut	BOOLEAN	R/O	BooleanRange R/O (true false)
state.rfOutLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
saveBootConfig	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device reset flag)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General device)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)
faults.08	ALARM	R/O	AlarmFlagRange R/O (RF upconverter DRO)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output overload)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output underload)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Internal M&C module)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Combiner module 1)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Upconv-Cond. module)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Internal communication)
internal.save	TEXT	R/O	StringRange R/O

9.452 Newtec-M6100

This page describes the device driver and the device window for the Newtec M6100 DVB-S2 Modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings.

- --- The TX interface settings.
- --- The transport stream multiplexer settings.
- --- Shows statistical values, error rates etc.
- --- The transport stream analyzer page
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the M6100.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ratePriority --- Selects if the modultor shall keep the symbol rate or the interface rate constant if e.g. modulation or fec rate is changed.
- extLoFreq --- The LO frequency of an external upconverter.
- extLoPolarity --- With NRM the driver assumes an LO frequency*below* the RF transmit frequency. With INV the LO frequency is assumed *above* the RF frequency.
- output10MHz --- Selects if a 10MHz reference clock shall be present at the L-Band output.
- onDeviceAlarm --- Specifies what to do on device alarms (IGNORE / TX-OFF)
- onIfcAlarm --- Specifies what to do on interface alarms (IGNORE / TX-OFF)

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwID	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.swID	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.pwrSupply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.ratePriority	CHOICE	SETUP	EnumRange (SYMBOL INTERFACE)
config.tsoipRate	CHOICE	SAVE SETUP	EnumRange (MANUAL AUTOMATIC)
config.extLoFreq	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.extLoPolarity	CHOICE	SAVE SETUP	EnumRange (NRM INV)
config.output10Mhz	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.onDeviceAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.onIfcAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.minOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.maxOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.readBackDelay	INTEGER		IntegerRange (0 .. 0)
config.hasProtectTSoIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
refClkSrc	CHOICE		EnumRange (INTERN EXTERN)
refClkFreq	CHOICE		EnumRange (1 MHz 2 MHz 5 MHz 10 MHz 20 MHz)
saveBootConfig	TEXT	NOPRESET	StringRange
unsavedChanges	BOOLEAN	R/O	BooleanRange R/O (true false)
dvbMode	CHOICE		EnumRange (DVB-S DVB-S2)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.0 .. 10.0)
tx.mod.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
tx.mod.occupiedBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.measuredRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.amplitudeEQ	INTEGER		IntegerRange (-7 .. 7)
tx.mod.dataRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.mod.type	CHOICE		EnumRange (QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mod.modcod	CHOICE		EnumRange (QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-6/7 QPSK-7/8 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mod.fec	CHOICE		EnumRange (1/2 1/3 1/4 2/3 2/5 3/4 3/5 4/5 5/6 8/9 9/10)

name	*type*	*flags*	*range*
tx.mod.modulation	CHOICE		EnumRange (QPSK 8PSK 16QAM 16APSK 32APSK)
tx.mod.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.insertPilots	CHOICE		EnumRange (ON OFF)
tx.mod.isi	INTEGER		IntegerRange (0 .. 255)
tx.mod.input	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.asi.in.inputSelect	CHOICE		EnumRange (NONE ASI 1 ASI 2 ASI 1 OR 2 ASI 1 BEFORE 2 ASI 2 BEFORE 1 PBRS GENERATOR)
protection	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.asi.in.activeInput	CHOICE	R/O	EnumRange R/O (OFF ASI 1 ASI 2)
tx.asi.in.position	CHOICE	R/O	EnumRange R/O (A B NONE)
tx.asi.in.inlineSplit	CHOICE		EnumRange (NONE 3DB SPLIT 6DB SPLIT)
tx.asi.in.measuredRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.asi.in.inputFraming	CHOICE		EnumRange (TS 188 TS 204)
tx.asi.in.itfSwitchCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.in.pbrsRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.asi.in.pbrsType	CHOICE		EnumRange (PBRS COUNTER)
tx.asi.in.pbrsPid	CHOICE		EnumRange (OFF ON)
tx.asi.in.pbrsPidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.in.pbrsDPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.in.pbrsNPackets	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
tx.asi.out.signalSelect	CHOICE		EnumRange (NONE MODULATOR INPUT ACTIVE INPUT NONACTIVE INPUT PBRS GENERATOR)
tx.asi.out.outputSelect	CHOICE		EnumRange (NONE ASI 1 ASI 2 ASI 1 AND 2)
tx.asi.out.measuredRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.asi.out.pbrsRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.asi.out.pbrsType	CHOICE		EnumRange (PBRS COUNTER)
tx.asi.out.pbrsPid	CHOICE		EnumRange (OFF ON)
tx.asi.out.pbrsPidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.out.pbrsDPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.out.pbrsNPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.mon.enable	CHOICE		EnumRange (OFF ON)
tx.asi.mon.pid	CHOICE		EnumRange (OFF ON)
tx.asi.mon.pidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.mon.state	TEXT	R/O	StringRange R/O
tx.asi.mon.syncSecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.mon.errorCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.mon.errorRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.tsoip.enable	CHOICE		EnumRange (OFF ON)
tx.tsoip.inputSelect	CHOICE		EnumRange (NONE DATA 1 DATA 2 DATA)
tx.tsoip.encapProtocol	CHOICE		EnumRange (UDP RTP RTP WITH FEC)

name	*type*	*flags*	*range*
tx.tsoip.addressType	CHOICE		EnumRange (UNICAST MULTICAST)
tx.tsoip.mcastAddress	TEXT		StringRange
tx.tsoip.udpPort	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.trafficProfile	CHOICE		EnumRange (VBR CBR)
tx.tsoip.maxTrafJitter	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.maxBuffDelay	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.tsoip.reset	TEXT		StringRange
tx.tsoip.mon.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.tsoip.mon.buffDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.maxBuffFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.minBuffFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.tsInCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.tsDropCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.tsOFlwCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpInCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpDropCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpRepairCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.mux.input.1.name	TEXT	R/O	StringRange R/O
tx.mux.input.1.enable	CHOICE		EnumRange (OFF ON)
tx.mux.input.2.name	TEXT	R/O	StringRange R/O
tx.mux.input.2.enable	CHOICE		EnumRange (OFF ON)
tx.mux.ra.enable	CHOICE		EnumRange (OFF ON)
tx.mux.ra.nullpDrop	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mux.ra.pcrRestamp	CHOICE		EnumRange (OFF ON)
tx.mux.sig.networkId	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.tsId	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.patRepRate	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.sdtRepRate	INTEGER		IntegerRange (0 .. 0)
tx.mux.cid.enable	CHOICE		EnumRange (OFF ON)
tx.mux.cid.descrTag	INTEGER		IntegerRange (0 .. 0)
tx.mux.cid.carrierId	TEXT		StringRange
tx.mux.cid.telnNo	TEXT		StringRange
tx.mux.cid.userInfo	TEXT		StringRange
tsanalyzer.enable	CHOICE		EnumRange (OFF ON)
tsanalyzer.reset	TEXT		StringRange
tsanalyzer.estimatedRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tsanalyzer.pidList	TEXT	R/O	StringRange R/O
tsanalyzer.1.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.1.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.1.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.1.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.1.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.2.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.2.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.2.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.2.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
tsanalyzer.2.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.3.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.3.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.3.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.3.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.3.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.4.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.4.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.4.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.4.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.4.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.5.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.5.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.5.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.5.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.5.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.6.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.6.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.6.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)

name	*type*	*flags*	*range*
tsanalyzer.6.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.6.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.7.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.7.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.7.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.7.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.7.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.8.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.8.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.8.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.8.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.8.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
tx.prot.tsoip.enable	CHOICE		EnumRange DIS (OFF ON)
tx.prot.tsoip.inputSelect	CHOICE		EnumRange DIS (TSoIP1 TSoIP2 TSoIP1-OR-TSoIP2 TSoIP1-BEFORE-TSoIP2 TSoIP2-BEFORE-TSoIP1)
tx.prot.tsoip.thresholdEnable	CHOICE		EnumRange DIS (OFF ON)
tx.prot.tsoip.thresholdMaxRate	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
tx.prot.tsoip.timeWindow	FLOAT		DoubleRange DIS (0.1 .. 5.0)

name	*type*	*flags*	*range*
tx.prot.tsoip.reset	TEXT	NOPRESET	StringRange DIS
tx.prot.tsoip.1.inputSelect	CHOICE		EnumRange DIS (NONE DATA 1 DATA 2 BOND ANY)
tx.prot.tsoip.2.inputSelect	CHOICE		EnumRange DIS (NONE DATA 1 DATA 2 BOND ANY)
tx.prot.tsoip.1.encapProtocol	CHOICE		EnumRange DIS (UDP RTP RTP WITH FEC)
tx.prot.tsoip.2.encapProtocol	CHOICE		EnumRange DIS (UDP RTP RTP WITH FEC)
tx.prot.tsoip.1.addressType	CHOICE		EnumRange DIS (UNICAST MULTICAST)
tx.prot.tsoip.2.addressType	CHOICE		EnumRange DIS (UNICAST MULTICAST)
tx.prot.tsoip.1.mcastAddress	TEXT		StringRange DIS
tx.prot.tsoip.2.mcastAddress	TEXT		StringRange DIS
tx.prot.tsoip.1.udpPort	INTEGER		IntegerRange DIS (0 .. 0)
tx.prot.tsoip.2.udpPort	INTEGER		IntegerRange DIS (0 .. 0)
tx.prot.tsoip.1.trafficProfile	CHOICE		EnumRange DIS (VBR CBR)
tx.prot.tsoip.2.trafficProfile	CHOICE		EnumRange DIS (VBR CBR)
tx.prot.tsoip.1.rateType	CHOICE		EnumRange DIS (USER DEFINED PCR)
tx.prot.tsoip.2.rateType	CHOICE		EnumRange DIS (USER DEFINED PCR)
tx.prot.tsoip.1.pcrAuto	CHOICE		EnumRange DIS (OFF ON)
tx.prot.tsoip.2.pcrAuto	CHOICE		EnumRange DIS (OFF ON)
tx.prot.tsoip.1.pcrPid	INTEGER		IntegerRange DIS (0 .. 8191)
tx.prot.tsoip.2.pcrPid	INTEGER		IntegerRange DIS (0 .. 8191)
tx.prot.tsoip.1.maxBuffDelay	INTEGER		IntegerRange DIS (0 .. 0)
tx.prot.tsoip.2.maxBuffDelay	INTEGER		IntegerRange DIS (0 .. 0)

name	*type*	*flags*	*range*
tx.prot.tsoip.1.dataRate	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
tx.prot.tsoip.2.dataRate	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
tx.prot.tsoip.mon.dataRate	FLOAT	R/O	DoubleRange R/O DIS (0.000000 .. 0.000000)
tx.prot.tsoip.activeInput	CHOICE	R/O	EnumRange R/O DIS (NONE TSoIP1 TSoIP2)
tx.prot.tsoip.1.mon.dataRate	FLOAT	R/O	DoubleRange R/O DIS (0.000000 .. 0.000000)
tx.prot.tsoip.2.mon.dataRate	FLOAT	R/O	DoubleRange R/O DIS (0.000000 .. 0.000000)
tx.prot.tsoip.1.mon.buffDelay	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.buffDelay	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.tsInCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.tsInCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.tsOutCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.tsOutCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.tsDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.tsDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.tsOFlwCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.tsOFlwCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
tx.prot.tsoip.1.mon.rtpInCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.rtpInCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.rtpDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.rtpDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.rtpRepairCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.rtpRepairCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.fecColIn	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.fecColIn	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.fecRowIn	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.fecRowIn	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.fecDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.fecDropCnt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.switchCount	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.maxBuffFill	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.maxBuffFill	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.minBuffFill	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
tx.prot.tsoip.2.mon.minBuffFill	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.1.mon.source	TEXT	R/O	StringRange R/O DIS
tx.prot.tsoip.2.mon.source	TEXT	R/O	StringRange R/O DIS
tx.prot.tsoip.1.mon.fecMode	TEXT	R/O	StringRange R/O DIS
tx.prot.tsoip.2.mon.fecMode	TEXT	R/O	StringRange R/O DIS
tx.prot.tsoip.1.mon.actPcrPid	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.prot.tsoip.2.mon.actPcrPid	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device general)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interface general)
faults.03	ALARM	R/O	AlarmFlagRange R/O (ASI general)
faults.04	ALARM	R/O	AlarmFlagRange R/O (ASI input signal)
faults.05	ALARM	R/O	AlarmFlagRange R/O (ASI input data)
faults.06	ALARM	R/O	AlarmFlagRange R/O (ASI buffer oflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TSoIP general)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TSoIP input data)
faults.09	ALARM	R/O	AlarmFlagRange R/O (TSoIP buffer oflow)
faults.10	ALARM	R/O	AlarmFlagRange R/O (TSoIP buffer uflow)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod general)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Mod input signal)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Mod buffer oflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Mod buffer uflow)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Mod synth)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Mod ref synth lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Mod lo1 lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Mod lo2 lock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TA ts sync)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TA ts sync byte)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TA ts pat)
faults.22	ALARM	R/O	AlarmFlagRange R/O (TA ts continuity)
faults.23	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr repetition)
faults.24	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr continuity)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr accuracy)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Protected 1 input data)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Protected 2 input data)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (Protected 1 buffer uflow)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Protected 2 buffer uflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Protected 1 buffer oflow)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Protected 2 buffer oflow)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Protected 1 rtp sync)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Protected 2 rtp sync)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Protected red)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Protected red degraded)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Protected input data)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Protected buffer uflow)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Protected buffer oflow)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Protected rtp sync)
internal.dvbMode	TEXT	R/O	StringRange R/O
internal.freq	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
internal.tx.on	TEXT	R/O	StringRange R/O
internal.type	TEXT	R/O	StringRange R/O

9.453 Newtec-MCX7000-ASIIN

This page describes the *Newtec-MCX7000-ASIIN* device driver and its device window. The *Newtec-MCX7000-ASIIN* device driver monitors and controls one ASI input configured in a

Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all ASI inputs of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'input' configuration parameter is used to select which of the physical ASI input ports in the Gateway this particular sat-nms device shall control. The driver supports up to 6 ASI interfaces.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- input --- Selects the physical ASI input in the gateway to control.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.input	CHOICE	SAVE SETUP	EnumRange (ASI-1 ASI-2 ASI-3 ASI-4 ASI-5 ASI-6)
inputEnable	CHOICE		EnumRange (OFF ON)
inputSplitter	CHOICE		EnumRange (NONE 3dB 6dB)
inputPacketSize	CHOICE		EnumRange (188 204)
inputBitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
operationState	TEXT	R/O	StringRange R/O
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input No Signal)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input No Data)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input Data Error)

9.454 Newtec-MCX7000-ASIOUT

This page describes the *Newtec-MCX7000-ASIOUT* device driver and its device window. The *Newtec-MCX7000-ASIOUT* device driver monitors and controls one ASI output configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all ASI outputs of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'output' configuration parameter is used to select which of the physical ASI output ports in the Gateway this particular sat-nms device shall control. The driver supports up to 6 ASI interfaces.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- output --- Selects the physical ASI output in the gateway to control.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.output	CHOICE	SAVE SETUP	EnumRange (ASI-1 ASI-2 ASI-3 ASI-4 ASI-5 ASI-6)
outputEnable	CHOICE		EnumRange (OFF ON)
outputBitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
operationState	TEXT	R/O	StringRange R/O
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (No Output)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output Overflow)

9.455 Newtec-MCX7000-BISS-Descr

This page describes the *Newtec-MCX7000-BISS-Descr* device driver and its device window. The *Newtec-MCX7000-BISS-Descr* device driver monitors and controls one BISS Descrambling instance configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all BISS instances of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'descrambler' configuration parameter is used to select which of the BISS instances in the Gateway this particular sat-nms device shall control. The software automatically evaluates the

number of available instances at the gateway after software startup and sets the available choices of the 'input' configuration parameter accordingly. The RESET button on the Maintenance device screen page can be used to force a re-read of the descrambler list, e.g. if a new BISS instance has been added via the Newtec Web GUI.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- descrambler --- Selects the BISS instance in the gateway to control. The software automatically evaluates the number of available BISS descrambling instances at the gateway after software startup and sets the available choices of this parameter accordingly. To force an update of the input list without restarting the software press the RESET button at the Maintenance page.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.buriedId	TEXT	R/O	StringRange R/O
config.descrambler	CHOICE	SAVE SETUP	EnumRange (1)
scrambling	CHOICE		EnumRange (OFF ON)
bissMode	CHOICE	SAVE	EnumRange (BISS-1 BISS-E)
bissEMode	CHOICE		EnumRange (BURIED INJECTED)
bissKey	TEXT	NOCOMPARE	StringRange
bissInjectedId	TEXT	NOCOMPARE	StringRange
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (PAT CRC)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PMT CRC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CAT CRC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (SW Refused)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Sig Overflow)

9.456 Newtec-MCX7000-Base

This page describes the *Newtec-MCX7000-Base* device driver and its device window. The

Newtec-MCX7000-Base device driver monitors and controls the base unit features and alarms of one Newtec MCX7000 Multi-carrier Satellite Gateway.

Beside this base unit multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the gateway All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The different available functional elements of the MCX7000 Multi-carrier Satellite Gateway are:

- Newtec-MCX7000-Demodulator monitors and controls one demodulator instance of a MCX7000.
- Newtec-MCX7000-IPIN monitors and controls one IP input of a MCX7000.
- Newtec-MCX7000-IPOUT monitors and controls one IP output of a MCX7000.
- Newtec-MCX7000-ASIIN monitors and controls one ASI input of a MCX7000.
- Newtec-MCX7000-ASIOUT monitors and controls one ASI output of a MCX7000.
- Newtec-MCX7000-BISS-Descr monitors and controls one BISS descrambler instance of a MCX7000.
- Newtec-MCX7000-Protection monitors and controls one TS protection instance of a MCX7000.

The driver allows to store the current units configuration as boot configuration. Saving a new boot configuration is only possible if the unit itself detects configurational changes compared to the current boot configuration. Additionally it allows a SW and HW reset of the whole Newtec MCX7000 Multi-carrier Satellite Gateway.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port

number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.uniqueId	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.deviceId	TEXT	R/O	StringRange R/O
info.hwId	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.swId	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
cpuLoad	TEXT	R/O	StringRange R/O
memoryUse	INTEGER	R/O	IntegerRange R/O (0 .. 0)
unsavedChanges	BOOLEAN	R/O	BooleanRange R/O (true false)
saveBootConfig	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
swReset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Boot configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Front Panel Com)
faults.06	ALARM	R/O	AlarmFlagRange R/O (NTP)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HW)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HW inventory)
faults.09	ALARM	R/O	AlarmFlagRange R/O (SW Upgrade)
faults.10	ALARM	R/O	AlarmFlagRange R/O (License invalid)
faults.11	ALARM	R/O	AlarmFlagRange R/O (License expire)
faults.12	ALARM	R/O	AlarmFlagRange R/O (License upgrade)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.457 Newtec-MCX7000-Demodulator

This page describes the *Newtec-MCX7000-Demodulator* device driver and its device window. The *Newtec-MCX7000-Demodulator* device driver monitors and controls one demodulator configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all demulators of the gateway. All devices referring to the same physical device must be

configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'demod' configuration parameter is used to select which of the defined demodulators in the Gateway this particular sat-nms device shall control.

The driver was tested with DVB-S, DVB-S2 (X) demodulation. Other modulation schemes and their specific parameter set were not implemented/tested. Depending on the selected modulation scheme specific monitoring parameters are available. For DVB-S the correct setting of modulation and fec is mandatory whereby this is automatically recognized using S2. Defining a physical layer scrambler signature only applies for S2/S2X.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
demod	Selects the IP output in the gateway to control. The

software automatically evaluates the number of available IP outputs at the gateway after software startup and sets the available choices of this parameter accordingly. To force an update of the output list without restarting the software press the RESET button at the Maintenance page. |

| spectrumSense | Defines the spectrum inversion sense of the demodulator. |

| marginThreshold | Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value is stored as a saved parameter on the mnc rather than as a parameter in the Demodulator. |

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 20000)

name	*type*	*flags*	*range*
config.demod	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.spectrumSense	CHOICE	SETUP	EnumRange (DIRECT INVERTED AUTO)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
enable	CHOICE		EnumRange (ON OFF)
input	CHOICE		EnumRange (1 2)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 TURBO AMDVBS AMCNBC DVBS2X AMCD)
streamMode	CHOICE		EnumRange (SINGLE MULTI)
frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
searchRange	INTEGER		IntegerRange (0 .. 0)
ifBandwidth	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rolloff	CHOICE		EnumRange (AUTO 5% 10% 15% 20% 25% 35%)
modCod	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 8PSK-2/3 8PSK-5/6 8PSK-8/9 16QAM-3/4 16QAM-7/8)
modulation	CHOICE		EnumRange (QPSK 8PSK 16QAM)
fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8)
scrambler	INTEGER		IntegerRange (0 .. 262141)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.level	CHOICE	R/O	EnumRange R/O (EXACT DOWN UP UNKNOWN)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.carrier	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.nonLinearPm	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.nonLinearAm	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.offset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.spectrumSense	CHOICE	R/O	EnumRange R/O (DIRECT INVERTED UNKNOWN)
meas.rolloff	CHOICE	R/O	EnumRange R/O (AUTO 5% 10% 15% 20% 25% 35% UNKNOWN)
state.viterbiBer	TEXT	R/O	StringRange R/O
state.ber	TEXT	R/O	StringRange R/O
state.qefMargin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pn	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.offsetToRef	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
frames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
framesError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
framesSlip	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
internal.enable	TEXT	R/O	StringRange R/O
internal.dvbMode	TEXT	R/O	StringRange R/O
internal.modfec	TEXT	R/O	StringRange R/O
internal.modCodDvbS	TEXT	R/O	StringRange R/O
internal.modCodDvbS2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Internal)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (High Input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PL Lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Demod Lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (LNB supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Overload)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.458 Newtec-MCX7000-IPIN

This page describes the *Newtec-MCX7000-IPIN* device driver and its device window. The *Newtec-MCX7000-ASIOUT* device driver monitors and controls one IP input configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all IP inputs of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'input' configuration parameter is used to select which of the IP inputs in the Gateway this particular sat-nms device shall control. The software automatically evaluates the number of available IP inputs at the gateway after software startup and sets the available choices of the 'input' configuration parameter accordingly. The RESET button on the Maintenance device screen page can be used to force a re-read of the input list, e.g. if a new IP input has been added via the Newtec Web GUI.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- input --- Selects the IP input in the gateway to control. The software automatically evaluates the number of available IP inputs at the gateway after software startup and sets the available choices of this parameter accordingly. To force an update of the input list without restarting the software press the RESET button at the Maintenance page.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.input	CHOICE	SAVE SETUP	EnumRange (1)
internal.enable	TEXT	R/O	StringRange R/O
inputEnable	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
inputProtocol	CHOICE		EnumRange (UDP RTP RTP-FEC)
srcPort	INTEGER		IntegerRange (1 .. 65535)
bufferDuration	INTEGER		IntegerRange (10 .. 1000)
bufferDurationMax	INTEGER		IntegerRange (10 .. 1000)
rateType	CHOICE		EnumRange (PCR USER)
bitrate	FLOAT		DoubleRange (0.000000 .. 0.000000)
pcrpid	INTEGER		IntegerRange (0 .. 8191)
loopType	CHOICE		EnumRange (NONE DVB NTC)
bindMode	CHOICE		EnumRange (UNBOUND MULTICAST)
autoPcr	CHOICE		EnumRange (ON OFF)
recoveryMode	CHOICE		EnumRange (SOURCE SEAMLESS)
recoveryWindow	INTEGER		IntegerRange (50 .. 900)
operationState	TEXT	R/O	StringRange R/O
meas.bitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
meas.offset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.bufferDuration	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.bufferDurationMin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.bufferDurationMax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.srcAddress	TEXT	R/O	StringRange R/O
meas.srcPort	TEXT	R/O	StringRange R/O
meas.fec	TEXT	R/O	StringRange R/O
meas.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.pcrpid	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meas.packetsNull	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.ip	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input data)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Buffer underflow)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Buffer overflow)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RTP Sync)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Buffer Delay)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Redundancy)

9.459 Newtec-MCX7000-IPOUT

This page describes the *Newtec-MCX7000-IPOUT* device driver and its device window. The *Newtec-MCX7000-IPOUT* device driver monitors and controls one IP output configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all IP outputs of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'output' configuration parameter is used to select which of the IP outputs in the Gateway this particular sat-nms device shall control. The software automatically evaluates the number of available IP outputs at the gateway after software startup and sets the available choices of the 'output' configuration parameter accordingly. The RESET button on the Maintenance device screen page can be used to force a re-read of the output list, e.g. if a new IP output has been added via the Newtec Web GUI.

The device driver cannot add or remove IP outputs, it simply reflects the actual configuration of the gateway. Adding or removing IP outputs and assigning them to physical network interfaces must be done with the Newtec Web GUI

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring an control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- output --- Selects the IP output in the gateway to control. The software automatically evaluates the number of available IP outputs at the gateway after software startup and sets the available choices of this parameter accordingly. To force an update of the output list without restarting the software press the RESET button at the Maintenance page.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.output	CHOICE	SAVE SETUP	EnumRange (1)
outputEnable	CHOICE		EnumRange (OFF ON)
tsProtocol	CHOICE		EnumRange (UDP RTP)
destPort	INTEGER		IntegerRange (1 .. 65535)
destIpAddress	TEXT		StringRange
ttl	INTEGER		IntegerRange (1 .. 255)
outputInterface	CHOICE	R/O	EnumRange R/O (DATA DATA1 DATA2)
tsPacketsInFrame	INTEGER		IntegerRange (1 .. 7)
flushScheduleTime	INTEGER		IntegerRange (0 .. 0)
outputBitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
dropCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
overflowCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
operationState	TEXT	R/O	StringRange R/O
packetRate	INTEGER		IntegerRange (0 .. 0)
flags.ip	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (No Output)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output Overflow)

9.460 Newtec-MCX7000-Modulator

This page describes the *Newtec-MCX7000-Modulator* device driver and it&#65533;s device window. The *Newtec-MCX7000-Modulator* device driver monitors and controls one Modulator configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all Modulators of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'modulator' configuration parameter is used to select which of the installed modulators in the Gateway this particular sat-nms device shall control.

The driver was prepared and tested with DVB-S and DVB-S2 (X) modulation. Other modulation types and their specific parameter set are not prepared

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device RF and Modulator monitoring and control parameters.
- --- The device Carrier ID parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
modulator	Selects the Modulator in the gateway to control.
spectrumSense	Defines the spectrum inversion sense of the Modulator.
ratePriority	Defines if the output rate should be set using symbol rate (SYMBOL) or data rate (INTERFACE).
onDeviceAlarm	Defines the transmit behaviour when an device alarm occurs.
onIfcAlarm	Defines the transmit behaviour when an interface alarm occurs.
transponderMode	Defines the modulator transponder mode setting.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 20000)
config.modulator	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.spectrumSense	CHOICE	SETUP	EnumRange (DIRECT INVERTED)

name	*type*	*flags*	*range*
config.ratePriority	CHOICE	SETUP	EnumRange (SYMBOL INTERFACE)
config.onDeviceAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.onIfcAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.transponderMode	CHOICE		EnumRange (SINGLE MULTI SINGLE-LIN)
tx.on	CHOICE		EnumRange (ON OFF)
tx.power	FLOAT		DoubleRange (-35.0 .. 10.0)
cwMode	CHOICE		EnumRange (CW MOD CLK/4 CLK/8 CLK/16 CID)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 TURBO AMDVBS AMCNBC DVBS2X AMCD)
streamMode	CHOICE		EnumRange (SINGLE MULTI)
frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
dataRate	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
rolloff	CHOICE		EnumRange (2% 5% 10% 15% 20% 25% 35%)
acmMode	CHOICE		EnumRange (ACM CCM)
modCod	CHOICE		EnumRange (QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 8PSK-2/3 8PSK-5/6 8PSK-8/9 16QAM-3/4 16QAM-7/8)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
pilots	CHOICE		EnumRange (ON OFF)
slicing	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
scramblerMode	CHOICE		EnumRange (STANDARD CONTINUOUS)
scrambler	INTEGER		IntegerRange (0 .. 262141)
rollOffSignal	CHOICE		EnumRange (STANDARD RESERVED)
cid.enable	CHOICE	NOPRESET	EnumRange (OFF ON)
cid.carrierId	TEXT	R/O	StringRange R/O
cid.sendGeo	CHOICE	NOPRESET	EnumRange (OFF ON)
cid.userInfo	TEXT	NOPRESET	StringRange
cid.longitude	FLOAT	NOPRESET	DoubleRange (-180.000000 .. 180.000000)
cid.latitude	FLOAT	NOPRESET	DoubleRange (-90.000000 .. 90.000000)
state.occupiedBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
state.transmit	TEXT	R/O	StringRange R/O
state.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.dvbMode	TEXT	R/O	StringRange R/O
internal.modCodDvbS	TEXT	R/O	StringRange R/O
internal.modCodDvbS2	TEXT	R/O	StringRange R/O
internal.modCodDvbS2x	TEXT	R/O	StringRange R/O
internal.modulator	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Buffer underflow)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Buffer overflow)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Frame Sync)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Calibration data)
faults.06	ALARM	R/O	AlarmFlagRange R/O (TS bitrate invalid)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Internal Failure)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Operation)

9.461 Newtec-MCX7000-Protection

This page describes the *Newtec-MCX7000-Protection* device driver and its device window. The *Newtec-MCX7000-Protection* device driver monitors and controls one TS redundancy protection configured in a Newtec MCX7000 Multi-carrier Satellite Gateway.

Multiple instances of the device may be configured in the sat-nms M&C to handle all TS redundancies of the gateway. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

The 'protection' configuration parameter is used to select which of the defined TS redundancies in the Gateway this particular sat-nms device shall control. The driver is prepared to support up to 6 redundancies on a single Newtec MCX7000 Multi-carrier Satellite Gateway.

The current position can be changed manually only by setting the protection to MANUAL and change the priority position afterwards. There are two protection modes available:

- Non-revertive: Switches only in case of a fault of the selected path.
- Revertive: Switches back to priority path when operational.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- protection --- Selects the ts redundancy in the gateway to control.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.protection	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
protectionEnable	CHOICE		EnumRange (OFF ON)
protection	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
protectionMode	CHOICE		EnumRange (ALWAYS SWITCH ONCE)
priority	CHOICE		EnumRange (A B)
input.a.type	TEXT	R/O	StringRange R/O
input.a.name	TEXT	R/O	StringRange R/O
input.b.type	TEXT	R/O	StringRange R/O
input.b.name	TEXT	R/O	StringRange R/O
operationState	TEXT	R/O	StringRange R/O
position	CHOICE	R/O	EnumRange R/O (A B NONE)
count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
inputBitrate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.enable	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (No Input)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.462 Newtec-MDM6100

This page describes the device driver and the device window for the Newtec MDM6100 DVB-S2 Modulator and Demodulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings.
- --- The demodulator settings.
- --- The TX interface settings.
- --- The RX interface settings.
- --- The transport stream multiplexer settings.
- --- Shows statistical values, error rates etc.
- --- The transport stream analyzer page

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the MDM6100.

parameter	***description***
address	The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
ratePriority	Selects if the modultor shall keep the symbol rate or the interface rate constant if e.g. modulation or fec rate is changed.
extLoFreq	The LO frequency of an external upconverter.
extLoPolarity	With NRM the driver assumes an LO frequency *below* the RF transmit frequency. With INV the LO frequency is assumed *above* the RF frequency.
output10MHz	Selects if a 10MHz reference clock shall be present at the L-Band output.
onDeviceAlarm	Specifies what to do on device alarms (IGNORE / TX-OFF)
minOutputLevel	Defines the minimum tx output power level (dBm) of the modulator as this cannot be read from the unit

parameter	*description*
maxOutputLevel	Defines the maximum tx output power level (dBm) of the modulator as this cannot be read from the unit
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances
rx.reference	Defines 10Mhz Reference signal
rx.lnbPower	Selects if the Demodulator shall apply LNB power to the input
rx.lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs
rx.lnbTone	Makes the Demodulator apply a 22 kHz tone to the LNB power when set to ON.
rx.marginThreshold	Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwID	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.swID	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.pwrSupply	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.ratePriority	CHOICE	SETUP	EnumRange (SYMBOL INTERFACE)
config.extLoFreq	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.extLoPolarity	CHOICE	SAVE SETUP	EnumRange (NRM INV)
config.output10Mhz	CHOICE	SETUP	EnumRange (OFF ON)
config.onDeviceAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.onIfcAlarm	CHOICE	SETUP	EnumRange (IGNORE TX OFF)
config.minOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.maxOutputLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.readBackDelay	INTEGER		IntegerRange (0 .. 0)
config.rx.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rx.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rx.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.rx.reference	CHOICE	SETUP	EnumRange (ON OFF)
config.rx.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.rx.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)

name	*type*	*flags*	*range*
config.rx.lnbTone	CHOICE	SETUP	EnumRange (OFF ON)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
refClkSrc	CHOICE		EnumRange (INTERN EXTERN)
refClkFreq	CHOICE		EnumRange (1 MHz 2 MHz 5 MHz 10 MHz 20 MHz)
saveBootConfig	TEXT	NOPRESET	StringRange
unsavedChanges	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.dvbMode	CHOICE		EnumRange (DVBS DVBS2)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.0 .. 10.0)
tx.mod.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
tx.mod.occupiedBw	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.measuredRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.amplitudeEQ	INTEGER		IntegerRange (-7 .. 7)
tx.mod.dataRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mod.modcod	CHOICE		EnumRange (QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-6/7 QPSK-7/8 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10 16QAM-3/4 16QAM-7/8 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10)
tx.mod.fec	CHOICE		EnumRange (1/2 1/3 1/4 2/3 2/5 3/4 3/5 4/5 5/6 8/9 9/10)
tx.mod.modulation	CHOICE		EnumRange (QPSK 8PSK 16QAM 16APSK 32APSK)
tx.mod.fecFrameType	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.insertPilots	CHOICE		EnumRange (ON OFF)
tx.mod.isi	INTEGER		IntegerRange (0 .. 255)
tx.mod.input	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.asi.in.inputSelect	CHOICE		EnumRange (NONE ASI 1 ASI 2 ASI 1 OR 2 ASI 1 BEFORE 2 ASI 2 BEFORE 1 PBRS GENERATOR)
protection	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.asi.in.activeInput	CHOICE	R/O	EnumRange R/O (OFF ASI 1 ASI 2)

name	*type*	*flags*	*range*
tx.asi.in.position	CHOICE	R/O	EnumRange R/O (A B NONE)
tx.asi.in.inlineSplit	CHOICE		EnumRange (NONE 3DB SPLIT 6DB SPLIT)
tx.asi.in.measuredRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.asi.in.inputFraming	CHOICE		EnumRange (TS 188 TS 204)
tx.asi.in.itfSwitchCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.in.pbrsRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.asi.in.pbrsType	CHOICE		EnumRange (PBRS COUNTER)
tx.asi.in.pbrsPid	CHOICE		EnumRange (OFF ON)
tx.asi.in.pbrsPidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.in.pbrsDPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.in.pbrsNPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.out.signalSelect	CHOICE		EnumRange (NONE MODULATOR INPUT ACTIVE INPUT NONACTIVE INPUT DEMODULATOR OUTPUT PBRS GENERATOR)
tx.asi.out.outputSelect	CHOICE		EnumRange (NONE ASI 3 ASI 4 ASI 3 AND 4)
tx.asi.out.measuredRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.asi.out.pbrsRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.asi.out.pbrsType	CHOICE		EnumRange (PBRS COUNTER)
tx.asi.out.pbrsPid	CHOICE		EnumRange (OFF ON)
tx.asi.out.pbrsPidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.out.pbrsDPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.out.pbrsNPackets	INTEGER		IntegerRange (0 .. 0)
tx.asi.mon.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.asi.mon.pid	CHOICE		EnumRange (OFF ON)
tx.asi.mon.pidVal	INTEGER		IntegerRange (0 .. 0)
tx.asi.mon.state	TEXT	R/O	StringRange R/O
tx.asi.mon.syncSecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.mon.errorCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.asi.mon.errorRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.tsoip.enable	CHOICE		EnumRange (OFF ON)
tx.tsoip.inputSelect	CHOICE		EnumRange (NONE DATA 1 DATA 2 DATA)
tx.tsoip.encapProtocol	CHOICE		EnumRange (UDP RTP RTP WITH FEC)
tx.tsoip.addressType	CHOICE		EnumRange (UNICAST MULTICAST)
tx.tsoip.mcastAddress	TEXT		StringRange
tx.tsoip.udpPort	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.trafficProfile	CHOICE		EnumRange (VBR CBR)
tx.tsoip.maxTrafJitter	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.maxBuffDelay	INTEGER		IntegerRange (0 .. 0)
tx.tsoip.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.tsoip.reset	TEXT		StringRange
tx.tsoip.mon.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.tsoip.mon.buffDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.maxBuffFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.minBuffFill	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.tsInCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.tsDropCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
tx.tsoip.mon.tsOFlwCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpInCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpDropCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.tsoip.mon.rtpRepairCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.mux.input.1.name	TEXT	R/O	StringRange R/O
tx.mux.input.1.enable	CHOICE		EnumRange (OFF ON)
tx.mux.input.2.name	TEXT	R/O	StringRange R/O
tx.mux.input.2.enable	CHOICE		EnumRange (OFF ON)
tx.mux.ra.enable	CHOICE		EnumRange (OFF ON)
tx.mux.ra.nullpDrop	CHOICE		EnumRange (OFF ON)
tx.mux.ra.pcrRestamp	CHOICE		EnumRange (OFF ON)
tx.mux.sig.networkId	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.tsId	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.patRepRate	INTEGER		IntegerRange (0 .. 0)
tx.mux.sig.sdtRepRate	INTEGER		IntegerRange (0 .. 0)
tx.mux.cid.enable	CHOICE		EnumRange (OFF ON)
tx.mux.cid.descrTag	INTEGER		IntegerRange (0 .. 0)
tx.mux.cid.carrierId	TEXT		StringRange
tx.mux.cid.telnNo	TEXT		StringRange
tx.mux.cid.userInfo	TEXT		StringRange
tsanalyzer.enable	CHOICE		EnumRange (OFF ON)
tsanalyzer.reset	TEXT		StringRange
tsanalyzer.estimatedRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tsanalyzer.pidList	TEXT	R/O	StringRange R/O
tsanalyzer.1.enable	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
tsanalyzer.1.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.1.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.1.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.1.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.2.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.2.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.2.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.2.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.2.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.3.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.3.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.3.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.3.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.3.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.4.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.4.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.4.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.4.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.4.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.5.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.5.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.5.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.5.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
tsanalyzer.5.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.6.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.6.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.6.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.6.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.6.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.7.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.7.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.7.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.7.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.7.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.8.enable	CHOICE	SAVE	EnumRange (OFF ON)
tsanalyzer.8.pid	INTEGER	SAVE	IntegerRange (0 .. 0)
tsanalyzer.8.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
tsanalyzer.8.continuityErr	BOOLEAN	R/O	BooleanRange R/O (true false)
tsanalyzer.8.transportErr	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
rx.on	CHOICE		EnumRange (ON OFF)
rx.dem.dvbMode	CHOICE		EnumRange (DVBS DVBS2 DVBS2X TURBO AMC-DVBS AMC-NBC AMC-DL DVBS2EXT)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.dem.symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
rx.dem.rollOff	CHOICE		EnumRange (AUTO 5% 10% 15% 20% 25% 35%)

name	*type*	*flags*	*range*
rx.input	CHOICE		EnumRange (RX-A RX-B IF)
rx.dem.spectrumInvert	CHOICE		EnumRange (AUTO NORMAL INVERT)
rx.dem.plsSignature	INTEGER		IntegerRange (0 .. 262141)
rx.dem.type	CHOICE		EnumRange (DUMMY-PL QPSK-NOFEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-AUTO QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 16APSK-2/3 16APSK-3/4 16APSK-4/5 16APSK-5/6 16APSK-8/9 16APSK-9/10 32APSK-3/4 32APSK-4/5 32APSK-5/6 32APSK-8/9 32APSK-9/10 16QAM-3/4 16QAM-7/8 16QAM-AUTO 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-AUTO 8PSK-5/6 8PSK-8/9 8PSK-9/10 AUTO)
rx.dem.streamMode	CHOICE		EnumRange (SINGLE MULTI)
rx.dem.dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
rx.dem.acqRange	INTEGER		IntegerRange (1 .. 5000)
rx.dem.plsKey	INTEGER		IntegerRange (0 .. 2000)
rx.dem.transponderMode	CHOICE		EnumRange (SCPC NON LINEAR MCPC SCPC LINEAR)
rx.flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.meter.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.carrierLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.frequencyOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.symbolRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
rx.meter.rollOff	TEXT	R/O	StringRange R/O
rx.meter.modcod	TEXT	R/O	StringRange R/O
rx.meter.frameType	TEXT	R/O	StringRange R/O
rx.meter.pilots	TEXT	R/O	StringRange R/O
rx.tsoip.enable	CHOICE		EnumRange (OFF ON)
rx.tsoip.protocol	CHOICE		EnumRange (UDP RTP RTP COLS ONLY RTP COLS AND ROWS)
rx.tsoip.port	INTEGER		IntegerRange (1 .. 65535)
rx.tsoip.address	TEXT		StringRange
rx.tsoip.ttl	INTEGER		IntegerRange (1 .. 255)
rx.tsoip.packetsFrame	INTEGER		IntegerRange (1 .. 7)
rx.tsoip.fecColumns	INTEGER		IntegerRange (0 .. 0)
rx.tsoip.fecRows	INTEGER		IntegerRange (0 .. 0)
rx.tsoip.schedule	INTEGER		IntegerRange (500 .. 5000)
rx.tsoip.mon.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rx.tsoip.mon.tsDropCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.tsoip.mon.tsOFlwCnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.tsoip.reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device general)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interface general)
faults.03	ALARM	R/O	AlarmFlagRange R/O (ASI general)
faults.04	ALARM	R/O	AlarmFlagRange R/O (ASI input signal)
faults.05	ALARM	R/O	AlarmFlagRange R/O (ASI input data)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (ASI buffer oflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TSoIP general)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TSoIP input data)
faults.09	ALARM	R/O	AlarmFlagRange R/O (TSoIP buffer oflow)
faults.10	ALARM	R/O	AlarmFlagRange R/O (TSoIP buffer uflow)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod general)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Mod input signal)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Mod buffer oflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Mod buffer uflow)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Mod synth)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Mod ref synth lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Mod lo1 lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Mod lo2 lock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TA ts sync)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TA ts sync byte)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TA ts pat)
faults.22	ALARM	R/O	AlarmFlagRange R/O (TA ts continuity)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr repetition)
faults.24	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr continuity)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TA ts pcr accuracy)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Demod general)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Demod internal)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Demod input saturated)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Demod pl lock)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Demod lock)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Demod lnb supply)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Demod overload)
faults.34	ALARM	R/O	AlarmFlagRange R/O (TSoIP Output)
faults.35	ALARM	R/O	AlarmFlagRange R/O (TSoIP Output Overflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)
internal.dvbMode	TEXT	R/O	StringRange R/O
internal.freq	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
internal.tx.on	TEXT	R/O	StringRange R/O
internal.type	TEXT	R/O	StringRange R/O
internal.lo	FLOAT	R/O	DoubleRange R/O (0 .. 0)

9.463 Newtec-NTC2080

This page describes the device driver and the device window for the Newtec NTC2080 DVB+8+16 Modulator. The driver also works with the NTC2062 and the equivalent Philips models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the NTC2080.

parameter	***description***
address	The modulator's packet address. Valid addresses are 49 .. 110, the address must be set even if the modulator is controlled via RS232.
variant	Select the unit variant 2080/2062 here. There some commands the 2062 does not understand, so selecting 2062 will prevent the driver from using these functions. A change to this selection does not become fully effective until the server is restarted.
freqMode	The NTC2080 driver automatically detects the converter module which is installed with the modulator. The transmit frequency either may be set in the L-band range (select freqMode=L-BAND) or in terms of converted RF, which may be satellite RF frequency or a 70MHz IF frequency, depending of the converter module used. To set the transmit frequency in the converter output frequency range, select freqMode=RF.
testData	You can make the modulator transmit a test data pattern instead of the data applied to it's input by setting this parameter to "PBRS".
channelEQ	You can set the modulator's channel equalizer with this parameter
amplitudeEQ	You can set the modulator's amplitude slope equalizer with this parameter

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the NTC2080. The proper setting is *9600,E,7,1* .

Communication protocol / Interface type

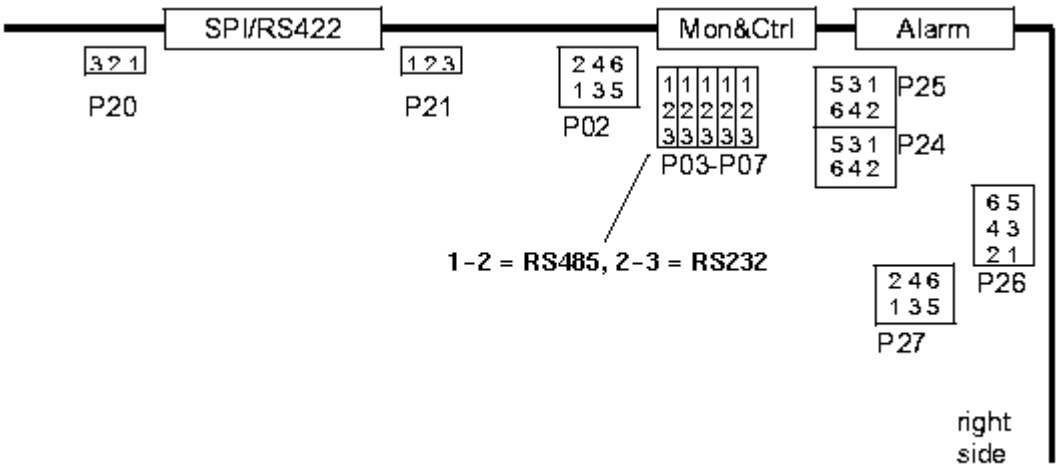
The NTC2080 device driver requires the *Newtec-Protocol* . You may use either a RS232 or a RS485 interface to control a NTC2080.

Installation

Before the sat-nms software will recognize the modulator, you have to set the communication parameters (device address, RS232/485 selection) at the unit's front panel. The procedure for the NTC2080 is as follows:

- Select the "SYSTEM MENU" using the UP/DOWN keys.
- Select the "Test functions ENABLED/DISABLED" item using the LEFT/RIGHT keys.
- If test functions are disabled, enter the password "2080" followed by ENTER to enable them.
- Select the RS485/RS232 item using the LEFT/RIGHT keys.
- Change to the interface type you intend to use.
- Revert the test functions to DISABLE.

With the NTC2060, the RS232/RS485 selection is done by setting five jumpers inside the unit. The figure below shows the location of these jumpers near the unit's back plane.



Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.variant	CHOICE	SAVE SETUP	EnumRange (NTC2080 NTC2062)
config.freqMode	CHOICE	SAVE SETUP	EnumRange (RF L-BAND)
config.testData	CHOICE	SETUP	EnumRange (OFF PRBS)
config.channelEQ	FLOAT	SETUP	DoubleRange DIS (0 .. 0)
config.amplitudeEQ	FLOAT	SETUP	DoubleRange (0 .. 0)
config.ifAtten	CHOICE	SETUP	EnumRange (ON OFF)
config.autoStore	CHOICE	SAVE SETUP	EnumRange (ON OFF)
info.revision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.deviceCaps	TEXT	R/O	StringRange R/O
info.hwDescription	TEXT	R/O	StringRange R/O
info.interfaceID	TEXT	R/O	StringRange R/O
info.converterID	TEXT	R/O	StringRange R/O
info.converterGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.refOscTemp	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
modemType	CHOICE		EnumRange (NTC SKYPLEX-SCPC SKYPLEX-TDMA DVB)
refClkSrc	CHOICE		EnumRange (INT EXT)
refClkFreq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
tx.mod.dataRate	INTEGER		IntegerRange (1000000 .. 110000000)
tx.mod.symbolRate	INTEGER		IntegerRange (1000000 .. 110000000)
tx.mod.type	CHOICE		EnumRange (QPSK 8PSK 16QUAM)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2 2/3 3/4 5/6 6/7 7/8 8/9)
tx.mod.rollOff	CHOICE		EnumRange (AUTO 0.35 0.25)
tx.mod.clockSource	CHOICE		EnumRange (EXTERN INTERN ADAPTED)
tx.mod.cwMode	CHOICE		EnumRange (MOD CW)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 204 NONE)
tx.ifc.hardware	CHOICE		EnumRange (DVB-SPI ECL RS422 ASI-A ASI-B ASI-OPT G703-2MB G703-6MB G703-8MB G703-17MB G703-34MB G703-45MB HSSI)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power on self test)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Device temperature)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (TX synthesizer)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TX HPA)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TX summary alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (10.0 MHz oscillator)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ext. ref. for 10.0 MHz)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF phase lock DRO)
faults.14	ALARM	R/O	AlarmFlagRange R/O (RF-head comm.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (100 MHz LO PLL)
faults.16	ALARM	R/O	AlarmFlagRange R/O (70 MHz LO PLL)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Clock PLL)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Data interface)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input framing)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Buffer over/underflow)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Converter summary)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Interface summary)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TX level)
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.minOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.maxOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.minRefOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxRefOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.minAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.maxAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.minChnEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.maxChnEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.TTy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.TTm	TEXT	R/O	StringRange R/O
internal.knowsRefOffs	TEXT	R/O	StringRange R/O
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)

9.464 Newtec-NTC2142

This page describes the device driver and the device window for the Newtec NTC 2142 Up and Downconverter with RCMP Protocol Version 1.0

It was tested with:

- HW version: 9.0 SW version: v1.07
- HW version: 10.0 SW version: v1.08

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- RX --- The receive parameters
- TX --- The transmit parameters
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
reference	Defines if internal or external reference is used.
autoStore	Switching this to YES makes the driver automatically save the device configuration into it's flash memory after every change. NO disables automatic saving, you may use the 'SAVE BOOT CONFIG' button in the sat-nms GUI to save the actual configuration explicitly.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.productId	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.hwId	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.autoStore	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.if.rxLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.if.txLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.txLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.int.txLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.if.frequency	CHOICE		EnumRange (70 MHz 140 MHz)
tx.polarity	CHOICE		EnumRange (DIRECT INVERTED)
tx.inputGain	FLOAT		DoubleRange (-20.0 .. 30.0)
tx.inputAttn	FLOAT		DoubleRange (-30.0 .. 20.0)
tx.inputBw	CHOICE		EnumRange (40 MHz 80 MHz)
tx.gain	FLOAT		DoubleRange (-23.0 .. 22.0)
tx.attn	FLOAT		DoubleRange (-22.0 .. 23.0)
tx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.int.on	CHOICE		EnumRange (OFF ON)
rx.frequency	FLOAT		DoubleRange (950.000000 .. 2150.000000)

name	*type*	*flags*	*range*
rx.gain	FLOAT		DoubleRange (0.0 .. 50.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 50.0)
rx.if.frequency	CHOICE		EnumRange (70 MHz 140 MHz)
rx.polarity	CHOICE		EnumRange (DIRECT INVERTED)
rx.if.on	CHOICE		EnumRange (ON OFF)
rx.outputBw	CHOICE		EnumRange (40 MHz 80 MHz)
reset	TEXT	NOPRESET	StringRange
clear	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Reset)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power on self test)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Incompatibilty)
faults.04	ALARM	R/O	AlarmFlagRange R/O (General)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Interface)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PS Voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (LO 1 Upconverter)
faults.10	ALARM	R/O	AlarmFlagRange R/O (LO 1 Downconverter)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LO 2 Upconverter)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LO 2 Downconverter)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input overload)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output overload)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output underload)
faults.26	ALARM	R/O	AlarmFlagRange R/O (M&C module)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Downconverter)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Architecture)

9.465 Newtec-NTC2180

This page describes the device driver and the device window for the Newtec NTC2180 DVB+8+16 Modulator with RCMP Protocol Version 1.0. This driver handles rf-on, internal and external l-band transmit separately, so you have full control of all the transmit switches.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the NTC2080.

parameter	*description*

parameter	***description***
address	The modulator's packet address. Valid addresses are 49 .. 110, the address must be set even if the modulator is controlled via RS232.
variant	Select the unit variant 2080/2062 here. There some commands the 2062 does not understand, so selecting 2062 will prevent the driver from using these functions. A change to this selection does not become fully effective until the server is restarted. Normally you need 2080 here for a 2180 series.
freqMode	The NTC2180 driver automatically detects the converter module which is installed with the modulator. The transmit frequency either may be set in the L-band range (select freqMode=L-BAND) or in terms of converted RF, which may be satellite RF frequency or a 70MHz IF frequency, depending of the converter module used. To set the transmit frequency in the converter output frequency range, select freqMode=RF.
testData	You can make the modulator transmit a test data pattern instead of the data applied to it's input by setting this parameter to "PBRS".
channelEQ	You can set the modulator's channel equalizer with this parameter
amplitudeEQ	You can set the modulator's amplitude slope equalizer with this parameter
ifAtten	You can enable or disable the modulator's IF attenuator with this parameter
Serial	Remember to set the line parameters for the serial interface connected to the NTC2080. The proper setting is *9600,E,7,1* .

Communication protocol / Interface type

The NTC2180 device driver requires the *Newtec-Protocol* . You may use either a RS232 or a RS485 interface to control a NTC2080. You have to configure RCMP Protocol Version 1.0 at the units front panel. If you have newer 2180 series modulator you can also try the NTC2277 drivers and RCMP Protocol Version 2.0.

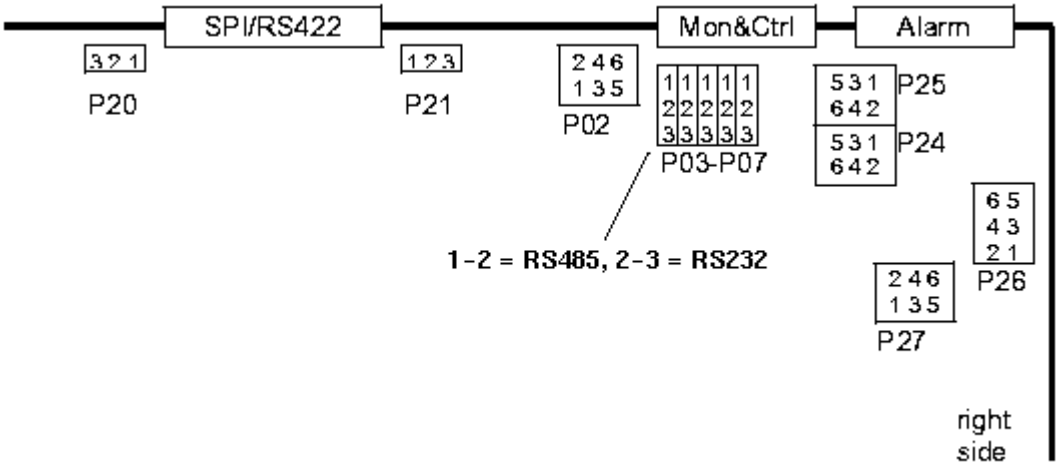
Installation

Before the sat-nms software will recognize the modulator, you have to set the communication parameters (device address, RS232/485 selection) at the unit's front panel. The procedure for the NTC2080 is as follows:

- Select the "SYSTEM MENU" using the UP/DOWN keys.
- Select the "Test functions ENABLED/DISABLED" item using the LEFT/RIGHT keys.
- If test functions are disabled, enter the password "2080" followed by ENTER to enable them.
- Select the RS485/RS232 item using the LEFT/RIGHT keys.

- Change to the interface type you intend to use.
- Revert the test functions to DISABLE.

With the NTC2060, the RS232/RS485 selection is done by setting five jumpers inside the unit. The figure below shows the location of these jumpers near the unit's back plane.



Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.variant	CHOICE	SAVE SETUP	EnumRange (NTC2080 NTC2062)
config.freqMode	CHOICE	SAVE SETUP	EnumRange (RF L-BAND)
config.testData	CHOICE	SETUP	EnumRange (OFF PRBS)

name	*type*	*flags*	*range*
config.channelEQ	FLOAT	SETUP	DoubleRange DIS (0 .. 0)
config.amplitudeEQ	FLOAT	SETUP	DoubleRange (0 .. 0)
config.ifAtten	CHOICE	SETUP	EnumRange (ON OFF)
config.autoStore	CHOICE	SAVE SETUP	EnumRange (ON OFF)
info.revision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.deviceCaps	TEXT	R/O	StringRange R/O
info.hwDescription	TEXT	R/O	StringRange R/O
info.interfaceID	TEXT	R/O	StringRange R/O
info.converterID	TEXT	R/O	StringRange R/O
info.converterGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.refOscTemp	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
modemType	CHOICE		EnumRange (NTC SKYPLEX-SCPC SKYPLEX-TDMA DVB)
refClkSrc	CHOICE		EnumRange (INT EXT)
refClkFreq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
tx.mod.dataRate	INTEGER		IntegerRange (1000000 .. 110000000)
tx.mod.symbolRate	INTEGER		IntegerRange (1000000 .. 110000000)
tx.mod.type	CHOICE		EnumRange (QPSK 8PSK 16QUAM)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2 2/3 3/4 5/6 6/7 7/8 8/9)
tx.mod.rollOff	CHOICE		EnumRange (AUTO 0.35 0.25)
tx.mod.clockSource	CHOICE		EnumRange (EXTERN INTERN ADAPTED)

name	*type*	*flags*	*range*
tx.mod.cwMode	CHOICE		EnumRange (MOD CW)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 204 NONE)
tx.ifc.hardware	CHOICE		EnumRange (DVB-SPI ECL RS422 ASI-A ASI-B ASI-OPT G703-2MB G703-6MB G703-8MB G703-17MB G703-34MB G703-45MB HSSI)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.power	FLOAT		DoubleRange (-35.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.intLBand	CHOICE		EnumRange (OFF ON)
tx.extLBand	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power on self test)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (TX synthesizer)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TX HPA)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TX summary alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (10.0 MHz oscillator)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ext. ref. for 10.0 MHz)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RF phase lock DRO)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (RF-head comm.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (100 MHz LO PLL)
faults.16	ALARM	R/O	AlarmFlagRange R/O (70 MHz LO PLL)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Clock PLL)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Data interface)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input framing)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Buffer over/underflow)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Converter summary)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Interface summary)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Incompatibility)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TX level)
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.minOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.maxOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.minRefOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxRefOffs	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.minAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.maxAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.minChnEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.maxChnEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.TTy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.TTm	TEXT	R/O	StringRange R/O
internal.knowsRefOffs	TEXT	R/O	StringRange R/O
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)

9.466 Newtec-NTC2277

This page describes the device driver and the device window for the Newtec NTC/2277 and NTC/2280 DVB Modulator with RCMP protocol version 2.0. Tested with IF-Band version (NTC/2277) with one modulator card installed and with the L-Band version (NTC/2280) without RF-Converter Module. Should work with other NTC/22xx and the corresponding NTC/21xx model as well.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the NTC2277.

parameter	*description*
address	The modulator's packet address. Valid addresses are 49 .. 110, the address must be set even if the modulator is controlled via RS232.

parameter	***description***
freqMode	Selects if the transmit frequency is set/displayed in terms of IF frequency (70/140MHz) or in terms of RF frequency (Ku-Band, L-Band, depends on the builtin upconverter). The setting must match the modulator's hardware configuration or the driver will state a communication fault.
testData	You can make the modulator transmit a test data pattern instead of the data applied to it's input by setting
amplitudeEQ	You can set the modulator's amplitude slope equalizer with this parameter
insertPilot	Enable/Disable DVB-S2 Physical Layer Pilot insertion
frameTypeFEC	Configures the DVB-S2 FEC-Frame type
gainControlMode	Configures the Gain Control Mode to MGC or AGC
monitorOutput	Select if the monitor output should also on or follows the tx signal
basebandInt	Select the baseband interface type (G.703 or ASI)
enableIfInput	Selects modulator's IF input ON/OFF (for chaining multiple modulators)
enableModConverter	Set to On if an internal converter module is available
Serial	Remember to set the line parameters for the serial interface connected to the NTC2277. The proper setting is *9600,E,7,1* .

Communication protocol / Interface type

The NTC2277 device driver requires the *Newtec-Protocol* . You may use either a RS232 or a RS485 interface to control a NTC2277. You can also use the Ethernet interface with this driver. To do this configure a network interface in the file *vlc.properties* with:

- *ttyN1.ip=aaa.bbb.ccc.ddd*
- *ttyN1.port=5933*

where aaa.bbb.ccc.ddd is the IP Address of the DVB-modulator.

Installation

Before the sat-nms software will recognize the modulator, you have to set the communication parameters (device address, RS232/485 selection or ip-settings) at the unit's front panel.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.freqMode	CHOICE	SAVE SETUP	EnumRange (IF RF)
config.testData	CHOICE	SETUP	EnumRange (OFF PRBS)
config.amplitudeEQ	INTEGER	SETUP	IntegerRange (0 .. 0)
config.gainControlMode	CHOICE	SETUP	EnumRange (MGC AGC)
config.monitorOutput	CHOICE	SETUP	EnumRange (ALWAYS ON FOLLOW TX)
config.basebandInt	CHOICE	SETUP	EnumRange (ASI G703)
config.enableIfInput	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.enableModConverter	CHOICE	SETUP	EnumRange (ON OFF)
config.autoStore	CHOICE	SAVE SETUP	EnumRange (ON OFF)
info.serialNo	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
info.hwversion	TEXT	R/O	StringRange R/O
info.hwDescription	TEXT	R/O	StringRange R/O
info.swDescription	TEXT	R/O	StringRange R/O
info.productID	TEXT	R/O	StringRange R/O
info.internalTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
modemType	CHOICE		EnumRange (NTC SKYPLEX-SCPC DVB-S DVB-S2)
refClkSrc	CHOICE		EnumRange (INTERN EXTERN)
tx.mod.dataRate	INTEGER	NOPRESET	IntegerRange (0 .. 0)
tx.mod.symbolRate	INTEGER		IntegerRange (0 .. 0)
tx.mod.type	CHOICE		EnumRange (QPSK-NONE QPSK-1/4 QPSK-1/3 QPSK-1/2 QPSK-2/5 QPSK-2/3 QPSK-3/5 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-8/9 QPSK-9/10 8PSK-2/3 8PSK-3/5 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10)
tx.mod.fec	CHOICE		EnumRange (NONE 1/4 1/3 1/2 2/5 2/3 3/5 3/4 4/5 5/6 6/7 7/8 8/9 9/10)
tx.mod.modulation	CHOICE		EnumRange (QPSK 8PSK)
tx.mod.rollOff	CHOICE		EnumRange (35% 25% 20%)
tx.mod.clockSource	CHOICE		EnumRange DIS (INTERN EXTERN)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD CLK/8 CLK/4 CLK/16)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 204 NONE)
tx.ifc.hardware	CHOICE		EnumRange (DVB-SPI RS422 HSSI int-BB ASI-A ASI-B ASI-OPT G703-2MB G703-6MB G703-8MB G703-17MB G703-34MB G703-45MB G703-155 MMF SMF OFF)
tx.ifc.asiSwitch	CHOICE	SETUP	EnumRange (MANUAL AUTO)
tx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
tx.power	FLOAT		DoubleRange (-35.0 .. -5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dvbs2.ACMMode	CHOICE		EnumRange DIS (CCM)
tx.mod.dvbs2.DFLMode	CHOICE		EnumRange DIS (auto manual)
tx.mod.dvbs2.isi	INTEGER		IntegerRange DIS (0 .. 255)
tx.mod.dvbs2.delayMode	CHOICE		EnumRange DIS (minimum programmed)
tx.mod.dvbs2.timingMode	CHOICE		EnumRange DIS (free running throttled)
tx.mod.dvbs2.physEfficiency	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
tx.mod.dvbs2.baseEfficiency	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
tx.mod.dvbs2.shortFrames	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
tx.mod.dvbs2.insertPilot	CHOICE		EnumRange DIS (ON OFF)
tx.mod.dvbs2.FECFrameType	CHOICE		EnumRange DIS (SHORT NORMAL)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Self test)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Incompatible settings)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Interface summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal interface)
faults.05	ALARM	R/O	AlarmFlagRange R/O (M&C)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Modulator)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Device temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Device architecture)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Power supply voltage)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Reference clock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input framing)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Input buffer overflow)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input buffer underflow)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ASI code violations)
faults.15	ALARM	R/O	AlarmFlagRange R/O (ASI optical signal)
faults.16	ALARM	R/O	AlarmFlagRange R/O (HSSI signal)
faults.17	ALARM	R/O	AlarmFlagRange R/O (LVDS signal)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Coax 1 Rx signal)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Coax 1 Rx code violations)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Coax 2 Rx signal)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Coax 2 Rx code violations)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Synthesiser)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Clock PLL)
faults.24	ALARM	R/O	AlarmFlagRange R/O (RF phase lock DRO)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Baseband frame sync)
faults.26	ALARM	R/O	AlarmFlagRange R/O (BISS summary)
faults.27	ALARM	R/O	AlarmFlagRange R/O (NCR inserter)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Internal upconverter)
faults.29	ALARM	R/O	AlarmFlagRange R/O (ODU communication)
faults.30	ALARM	R/O	AlarmFlagRange R/O (ODU architecture)
faults.31	ALARM	R/O	AlarmFlagRange R/O (ODU summary)
internal.minOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.maxOutput	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.minAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.maxAmpEQ	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.TTm	CHOICE		EnumRange (OFF ON)
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.type	TEXT	R/O	StringRange R/O

9.467 Newtec-PEP

This page describes the device driver and the device window for the Newtec PEP . The driver accesses the device via HTTP do get the overall status information. To monitor the interface state and other common parameters use the MIB2-Interface-Monitor device . It works for a single PEP or for a 1+1 redundant configuration.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Status information
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the AZ110.

- *parameter --- description*
- address --- The IP address of the HTTP interface and the port number.*aaa.bbb.ccc.ddd:pppp*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- config.host1 --- For redundant config only: hostname of the first PEP
- config.host2 --- For redundant config only: hostname of the second PEP

1+1 Redundancy

If two PEP's are configured in a 1+1 redundancy, they share a common IP address. Only the running (master) PEP will provide the TelliNet Webinterface on this address (normally on port 8000)

Configure only one Newtec-PEP device in the sat-nms software and set the address to the common IP address and configure the hostnames of both units. The driver will compare both names against the name of the active PEP and you can access this information via the parameter *active* .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.s	TEXT	R/O	StringRange R/O
config.host1	TEXT	SAVE SETUP	StringRange
config.host2	TEXT	SAVE SETUP	StringRange
hostname	TEXT	R/O	StringRange R/O
status	TEXT		StringRange
date	TEXT		StringRange
active	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.468 Newtec-USS0202

This page describes the device driver and the device window for the Newtec 1+1 Universal redundancy controller

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The monitoring parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwName	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.modulator0IP	TEXT	R/O	StringRange R/O
info.modulator0Name	TEXT	R/O	StringRange R/O
info.modulator1IP	TEXT	R/O	StringRange R/O
info.modulator1Name	TEXT	R/O	StringRange R/O
info.autoswitch	TEXT	R/O	StringRange R/O
info.switchback	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
location	CHOICE		EnumRange (A B)
switchMode	CHOICE		EnumRange (AUTO MANUAL)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Bad Device Type)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Bad USS configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (No USS configuration)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Physical Switch Failed)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Save USS Config Failed)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Access Failed)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Module Access Failed)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PSU A Failed)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU A Failed)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Internal Error)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Bootstrap)
faults.12	ALARM	R/O	AlarmFlagRange R/O (System Time Not Set)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Copy Failed)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Degraded State)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Device Alarm Link Lost)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Device Monitor Consistency Link Lost)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Device State Inconsistency)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Excessive AutoSwitch)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Learn Device Configuration Failed)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Physical Switch Failed)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (Spare Device Alarm)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch Ability Lost)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch Failed)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Warning! Main Disabled)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Warning! Spare Disabled)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Warning! Spare Interface Alarm)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Warning! Switchback Not Active)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Warning! Switched Disabled)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Mod0 Alarm)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Mod1 Alarm)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Mod0 Interface Alarm)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Mod0 Interface Alarm)

9.469 Newtec-USS0212

This page describes the device driver and the device window for the Newtec USS0212 protection switch via SNMP.

It is used controlling a single 1:1 redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter --- description

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwName	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.modulator0IP	TEXT	R/O	StringRange R/O
info.modulator0Name	TEXT	R/O	StringRange R/O
info.modulator1IP	TEXT	R/O	StringRange R/O
info.modulator1Name	TEXT	R/O	StringRange R/O
info.autoswitch	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.switchback	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (A B)
switchMode	CHOICE		EnumRange (AUTO MANUAL)
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Bad Device Type)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Bad USS configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (MGMT 1 Lost)
faults.04	ALARM	R/O	AlarmFlagRange R/O (MGMT 2 Lost)
faults.05	ALARM	R/O	AlarmFlagRange R/O (No USS configuration)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Physical Switch)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Save USS Config)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Access)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module Access)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU A)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU B)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Internal Error)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Bootstrap)
faults.14	ALARM	R/O	AlarmFlagRange R/O (System Time)
faults.15	ALARM	R/O	AlarmFlagRange R/O (G1 Copy Failed)
faults.16	ALARM	R/O	AlarmFlagRange R/O (G1 Degraded State)
faults.17	ALARM	R/O	AlarmFlagRange R/O (G1 Device Alarm Link)
faults.18	ALARM	R/O	AlarmFlagRange R/O (G1 Device Mon Consistency Link)
faults.19	ALARM	R/O	AlarmFlagRange R/O (G1 Device State Inconsistency)
faults.20	ALARM	R/O	AlarmFlagRange R/O (G1 Excessive AutoSwitch)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (G1 Learn Device Config)
faults.22	ALARM	R/O	AlarmFlagRange R/O (G1 Physical Switch)
faults.23	ALARM	R/O	AlarmFlagRange R/O (G1 Spare Device)
faults.24	ALARM	R/O	AlarmFlagRange R/O (G1 Switch Ability)
faults.25	ALARM	R/O	AlarmFlagRange R/O (G1 Switch)
faults.26	ALARM	R/O	AlarmFlagRange R/O (G1 Spare Interface)
faults.27	ALARM	R/O	AlarmFlagRange R/O (G1 Switchback Not Active)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Invalid position)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Mod0 Device)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Mod1 Device)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Mod0 Interface)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Mod1 Interface)

9.470 Noise-Source-Measurement

The Noise-Source-Measurement logical device controls the measurement procedure for a noise source calibration using a 'Decimator-Spectrum-Analyzer'. For a measurement the device sets the spectrum analyzer parameters and the signal path. It switches off the noise source for the first measurement and takes one spectrum measurement. Then it switches on the noise source and takes a second spectrum measurement.

The results of the two measurements are saved as csv-files (tables with two columns:frequency and level).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Monitors and controls the measurement.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which configure the behaviour of the device.

parameter	***description***
spectrAnalyzer	The device name of the spectrum analyzer to use for the measurement. Actually only devices of the type 'Decimator-Spectrum-Analyzer' are supported.
noiseSrcId	The message ID of the parameter which switches the noise source ON or OFF.
directory	The directory where to save the measurement result files containing the spectrum data. May be left empty (files are saved in the home directory of the sat-nms software), relative or absolute (starting with a '/' character).

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.noiseSrcId	TEXT	SAVE SETUP	StringRange
config.spectrAnalyzer	TEXT	SAVE SETUP	StringRange
config.directory	TEXT	SAVE SETUP	StringRange
preMeasSwitchId.1	TEXT	SAVE	StringRange
preMeasSwitchId.2	TEXT	SAVE	StringRange
preMeasSwitchId.3	TEXT	SAVE	StringRange
preMeasSwitchId.4	TEXT	SAVE	StringRange
preMeasSwitchVal.1	TEXT	SAVE	StringRange
preMeasSwitchVal.2	TEXT	SAVE	StringRange
preMeasSwitchVal.3	TEXT	SAVE	StringRange
preMeasSwitchVal.4	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
start	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
rbw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
vbw	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
avg	INTEGER	SAVE	IntegerRange (1 .. 255)
pdfilter	CHOICE	SAVE	EnumRange (VBW AVG)
measure	TEXT	NOPRESET	StringRange
progress	TEXT	R/O	StringRange R/O

9.471 NovelSat-NS2000

This page describes the device driver and the device window for the NovelSat NS2000 DVB-S2 and NS3 demodulator.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The main receive parameters, including frequency and channel parameters.
- --- Receive quality and other parameters measured by the device
- --- The ASI and Ethernet output parameters
- --- Faults and fault mask
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for a NS2000 may be *10.0.0.1 public private* .

- 22kHz --- Turns the 22 kHz signal in the LNB supply on or off. Only effective if 22kHzAuto is set OFF.
- 22kHzAuto --- Switching this on, makes the driver controlling the 22kHz signal with the receive frequency. For frequencies above the "band edge" 22kHz gets switched on.
- lnbVoltage --- Selects the LNB voltage (13V/18V) or switches it OFF.
- lnbPolSwitch --- Selects if the LNB voltage shall be permanently set to the the value selected by "lnbVoltage" or if the voltage shall be selected from the "polarization" parameter of the driver.
- loFreqHigh --- The high band LO frequency.
- loFreqLow --- The low band LO frequency.
- bandEdge --- The band edge frequency.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.swVersion	TEXT	R/O	StringRange R/O
info.firmwareVersion	TEXT	R/O	StringRange R/O
info.osVersion	TEXT	R/O	StringRange R/O
info.fsVersion	TEXT	R/O	StringRange R/O
info.fpgaTemperature	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.boardTemperature	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
config.loFreqLow	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
config.bandEdge	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
config.ch1.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.ch2.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.22kHz	CHOICE	R/O	EnumRange R/O (OFF ON)
internal.lnbVoltage	CHOICE	R/O	EnumRange R/O (OFF 13V 18V)
internal.lnbCtrl	TEXT	R/O	StringRange R/O
dvbmode	CHOICE		EnumRange (DVBS DSNG DVBS2 NS3 NS4 n/a)
frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	CHOICE		EnumRange (Y X)
rolloff	CHOICE		EnumRange (2% 5% 10% 15% 20% 25% 35% n/a)
dualChannelMode	CHOICE		EnumRange (SINGLE DUAL)
channel1.isiFrom	INTEGER		IntegerRange (0 .. 255)
channel1.isiTo	INTEGER		IntegerRange (0 .. 255)
channel2.isiFrom	INTEGER		IntegerRange (0 .. 255)
channel2.isiTo	INTEGER		IntegerRange (0 .. 255)
ns4NlcMode	CHOICE		EnumRange (OFF ON N/A)
nonLinearMode	CHOICE		EnumRange (OFF ON)
acmMode	CHOICE		EnumRange (OFF ON)
acmServerAddress	TEXT		StringRange
acmMargin	FLOAT		DoubleRange (0.00 .. 0.00)
autoCcmMode	CHOICE		EnumRange (OFF ON)
goldenSequence	INTEGER		IntegerRange (0 .. 262141)
asi.1.source	CHOICE		EnumRange (DISABLE CHANNEL-1 CHANNEL-2)
asi.1.dataInvert	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
asi.1.dataRateMode	CHOICE		EnumRange (AUTO MANUAL)
asi.1.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
asi.1.rateAccuracy	FLOAT		DoubleRange (0.00 .. 0.00)
asi.2.source	CHOICE		EnumRange (DISABLE CHANNEL-1 CHANNEL-2)
asi.2.dataInvert	CHOICE		EnumRange (OFF ON)
asi.2.dataRateMode	CHOICE		EnumRange (AUTO MANUAL)
asi.2.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
asi.2.rateAccuracy	FLOAT		DoubleRange (0.00 .. 0.00)
eth.1.source	CHOICE		EnumRange (DISABLE CHANNEL-1 CHANNEL-2)
eth.1.autoNegotiation	CHOICE		EnumRange (OFF ON)
eth.1.speed	CHOICE		EnumRange (10M 100M)
eth.1.duplex	CHOICE		EnumRange (HALF FULL)
eth.1.encapsulation	CHOICE		EnumRange (RAW NSPE GSE ULE)
eth.1.pid	INTEGER		IntegerRange (0 .. 8191)
eth.2.source	CHOICE		EnumRange (DISABLE CHANNEL-1 CHANNEL-2)
eth.2.autoNegotiation	CHOICE		EnumRange (OFF ON)
eth.2.speed	CHOICE		EnumRange (10M 100M)
eth.2.duplex	CHOICE		EnumRange (HALF FULL)
eth.2.encapsulation	CHOICE		EnumRange (RAW NSPE GSE ULE)
eth.2.pid	INTEGER		IntegerRange (0 .. 8191)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.compositeLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pilotSnr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
flags.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ch1.linkMargin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch1.snr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch1.fer	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)
flags.ch1.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ch2.linkMargin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch2.snr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch2.fer	FLOAT	R/O	DoubleRange R/O (0.00E+00 .. 0.00E+00)
flags.ch2.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.dvbmode	TEXT	R/O	StringRange R/O
state.frequency	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequencyOffset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.symbolRateOffset	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.rolloff	TEXT	R/O	StringRange R/O
state.ch1.dataRate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch1.modulation	TEXT	R/O	StringRange R/O
state.ch1.fec	TEXT	R/O	StringRange R/O
state.ch1.pilot	TEXT	R/O	StringRange R/O
state.ch1.frameSize	TEXT	R/O	StringRange R/O
state.ch2.dataRate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ch2.modulation	TEXT	R/O	StringRange R/O
state.ch2.fec	TEXT	R/O	StringRange R/O
state.ch2.pilot	TEXT	R/O	StringRange R/O
state.ch2.frameSize	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Channel 1 lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel 2 lock)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Channel1 Margin)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Channel2 Margin)

9.472 Novella-Blockdownconverter

This page describes the device driver and the device window for Novella Blockdownconverter via SNMP.

The driver was tested with the following models:

- BD501S SW: 95.7204a/1.00
- BD511S SW: 95.7204a/1.00
- BD702S SW: 95.7204a/1.00

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The device control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ctrlMode --- Defines the control mode REMOTE/LOCAL
- channels --- Defines number of available channels

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.location	TEXT	R/O	StringRange R/O
info.mainSwVersion	TEXT	R/O	StringRange R/O
info.netSwVersion	TEXT	R/O	StringRange R/O
info.configVersion	TEXT	R/O	StringRange R/O
config.ctrlMode	CHOICE	SETUP	EnumRange (REMOTE LOCAL)

name	*type*	*flags*	*range*
config.channels	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn.3	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn.4	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fault 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fault 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fault 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fault 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fault 5)

9.473 Novella-Blockupconverter

This page describes the device driver and the device window for Novella Blockupconverter via SNMP.

The driver was tested with the following models:

- BU501 SW: 95.7204a/1.00
- BU731 SW: 95.7204a/1.00

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The device control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ctrlMode --- Defines the control mode REMOTE/LOCAL
- channels --- Defines number of available channels

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.location	TEXT	R/O	StringRange R/O
info.mainSwVersion	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.netSwVersion	TEXT	R/O	StringRange R/O
info.configVersion	TEXT	R/O	StringRange R/O
config.ctrlMode	CHOICE	SETUP	EnumRange (REMOTE LOCAL)
config.channels	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn.3	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn.4	FLOAT		DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fault 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fault 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fault 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fault 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fault 5)

9.474 Novella-RU03

This page describes the device driver and the device window for Novella RU03 1:1 Redundancy Controller for Blockup- and downconverters via SNMP.

It was tested with SW version 97-6055/V1.0

The Novella RU04 covers units in 2:1 redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
ctrlMode	Defines the control mode REMOTE/LOCAL

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.siteDescription	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.description	TEXT	R/O	StringRange R/O
config.ctrlMode	CHOICE	SETUP	EnumRange (REMOTE LOCAL)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
position	CHOICE	R/O	EnumRange R/O (A B)
location	CHOICE		EnumRange (1 2)
protection	CHOICE		EnumRange (AUTO MANUAL)
internal.f1	TEXT	R/O	StringRange R/O
internal.f2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.475 Novella-RU04

This page describes the device driver and the device window for Novella RU04 2:1 Redundancy Controller for Blockup- and downconverters via SNMP.

It was tested with SW version 97-6050/V1.0

The Novella RU03 covers units in 2:1 redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- ctrlMode --- Defines the control mode REMOTE/LOCAL

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.description	TEXT	R/O	StringRange R/O
config.ctrlMode	CHOICE	SETUP	EnumRange (REMOTE LOCAL)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
internal.f1	TEXT	R/O	StringRange R/O
internal.f2	TEXT	R/O	StringRange R/O
internal.f3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unit 3)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.476 Novella-Tracking-Receiver

This page describes the device driver and the device window for the Novella tracking or beacon receiver (L,C,X and/or Ku-Band).

The driver detects the model name and adapts the frequency range.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Calibration

"offset" is the calibration value to calculate the dB from the DC voltage steps:

1. calibrate the unit on the front panel with "set DB"
2. calibrate the sat-nms device driver by changing offset parameter. until the signal level displayed by the M&C system and the frontpanel matchs. Scale is 2 for all known models

LO Frequency

The conversion (LO) frequency is handled by the drivers and not by the Novella Receiver. Common values for this frequency are 10000 MHz or 11300 MHz for consumer. Ku-Band antennas. Set this value to zero if you intend to set the receive frequency in terms of L-band frequency.. If the LO is above above the receive frequency (e.g. for a C-Band. application) enter

the LO frequency as a negative value

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.powerHex	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.powerDC	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.controlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.scale	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.offset	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.LOFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.test	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.477 Novotronik-GTA5200

This page describes the device driver and the device window for a Novotronik Multicoupler GT5200 series.

It was tested with the following models:

- GTA5202.16,
- GTA5203.16,
- GTA5201.4.

The driver is intended for monitoring only purpose without its own parameters page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTA5200 do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Voltage 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Voltage 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal System)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)

9.478 Novotronik-GTS0118

This page describes the device driver and the device window for a Novotronik 8x1 Switch GTS0118.

Please note: This driver is for GTS0118 wird 8 switches installed. Please use GTS0752 for devices with 9 switches.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS0118 do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8)
position.01	CHOICE	R/O	EnumRange R/O (A B)
position.02	CHOICE	R/O	EnumRange R/O (A B)
position.03	CHOICE	R/O	EnumRange R/O (A B)
position.04	CHOICE	R/O	EnumRange R/O (A B)
position.05	CHOICE	R/O	EnumRange R/O (A B)
position.06	CHOICE	R/O	EnumRange R/O (A B)
position.07	CHOICE	R/O	EnumRange R/O (A B)
position.08	CHOICE	R/O	EnumRange R/O (A B)
s01.position	CHOICE		EnumRange (A B)
s02.position	CHOICE		EnumRange (A B)
s03.position	CHOICE		EnumRange (A B)
s04.position	CHOICE		EnumRange (A B)
s05.position	CHOICE		EnumRange (A B)
s06.position	CHOICE		EnumRange (A B)
s07.position	CHOICE		EnumRange (A B)
s08.position	CHOICE		EnumRange (A B)
wgstat.01	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.02	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.03	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.04	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.05	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
wgstat.06	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.07	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.08	HEX	R/O	HexRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.479 Novotronik-GTS0752

This page describes the device driver and the device window for a Novotronik 9x1 Switch GTS0752.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS0752 do not need any configuration parameters except the settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9)
position.01	CHOICE	R/O	EnumRange R/O (A B)
position.02	CHOICE	R/O	EnumRange R/O (A B)
position.03	CHOICE	R/O	EnumRange R/O (A B)
position.04	CHOICE	R/O	EnumRange R/O (A B)
position.05	CHOICE	R/O	EnumRange R/O (A B)
position.06	CHOICE	R/O	EnumRange R/O (A B)
position.07	CHOICE	R/O	EnumRange R/O (A B)
position.08	CHOICE	R/O	EnumRange R/O (A B)
position.09	CHOICE	R/O	EnumRange R/O (A B)
s01.position	CHOICE		EnumRange (A B)
s02.position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
s03.position	CHOICE		EnumRange (A B)
s04.position	CHOICE		EnumRange (A B)
s05.position	CHOICE		EnumRange (A B)
s06.position	CHOICE		EnumRange (A B)
s07.position	CHOICE		EnumRange (A B)
s08.position	CHOICE		EnumRange (A B)
s09.position	CHOICE		EnumRange (A B)
wgstat.01	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.02	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.03	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.04	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.05	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.06	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.07	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.08	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.09	HEX	R/O	HexRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.480 Novotronik-GTS0752A

This page describes the device driver and the device window for a Novotronik 9x1 Switch GTS0752A.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS0752A do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
position	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9)
position.01	CHOICE	R/O	EnumRange R/O (A B)
position.02	CHOICE	R/O	EnumRange R/O (A B)
position.03	CHOICE	R/O	EnumRange R/O (A B)
position.04	CHOICE	R/O	EnumRange R/O (A B)
position.05	CHOICE	R/O	EnumRange R/O (A B)
position.06	CHOICE	R/O	EnumRange R/O (A B)
position.07	CHOICE	R/O	EnumRange R/O (A B)
position.08	CHOICE	R/O	EnumRange R/O (A B)
position.09	CHOICE	R/O	EnumRange R/O (A B)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.481 Novotronik-GTS4600

This page describes the device driver and the device window for a Novotronik 8x1 Switch from type GTS4600 made as specified by ASTRA Platform Services GmbH.

The switch has 8 inputs, possible position setting values are 0,1,2...8, position 0 means that no input is connected to the output and all ports are terminated with proper matching impedances.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS4600 do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
position	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)

9.482 Novotronik-GTS5022

This page describes the device driver and the device window for a Novotronik 4x1 Switch from type GTS5022.

The switch has 4 ports, possible position setting values are 0,1,2...4, position 0 means that no input is connected to the output and all ports are terminated with proper matching impedances.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS5022 do not need any configuration parameters except the settings for the serial interface

Remote control

Remote control is available via TCP Port 9998 with the "Terminal-LF-LFCR" ASCII protocol. Remember to switch to Remote Mode (Push button on Front Panel) to allow remote control of switch positions.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.partNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
position	CHOICE		EnumRange (0 1 2 3 4)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input 12VDC)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal 12VDC)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Internal 5VDC)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch board)

9.483 Novotronik-GTS5460

This page describes the device driver and the device window for a Novotronik BDC Switching Unit GTS5460.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching 1-4
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS5460 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.switchLabel.11	TEXT	SAVE SETUP	StringRange
config.switchLabel.12	TEXT	SAVE SETUP	StringRange
config.switchLabel.13	TEXT	SAVE SETUP	StringRange
config.switchLabel.14	TEXT	SAVE SETUP	StringRange
position.11	CHOICE		EnumRange (A B)
position.12	CHOICE		EnumRange (A B)
position.13	CHOICE		EnumRange (A B)
position.14	CHOICE		EnumRange (A B)
wgstat.11	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.12	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.13	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.14	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.11	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
wgcmd.12	HEX		HexRange (0 .. 0)
wgcmd.13	HEX		HexRange (0 .. 0)
wgcmd.14	HEX		HexRange (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

9.484 Novotronik-GTS5852

This page describes the device driver and the device window for a Novotronik 8x1 Switch from type GTS5852.

The switch has 8 inputs, possible position setting values are 0,1,2...8, position 0 means that no input is connected to the output and all ports are terminated with proper matching impedances.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS5852 do not need any configuration parameters except the settings for IP address and SNMP communities

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for may be *10.0.0.1 public private* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
position	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12)
faults.01	ALARM	R/O	AlarmFlagRange R/O (CRK 2000286)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch Board)
faults.04	ALARM	R/O	AlarmFlagRange R/O (K1 read back)
faults.05	ALARM	R/O	AlarmFlagRange R/O (K2 read back)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (K3 read back)
faults.07	ALARM	R/O	AlarmFlagRange R/O (24VDC-1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (24VDC-2)

9.485 Novotronik-GTS6300

This page describes the device driver and the device window for a Novotronik LNA Switching Unit GTS6300.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch parameters.
- --- Switch readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS6300 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
config.controlMode	Switches the device control mode between local and remote.
config.dcPs1	Switches DC power supply 1.
config.dcPs2	Switches DC power supply 2.
config.dcPs3	Switches DC power supply 3.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.controlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.dcPs1	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs2	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs3	CHOICE	SETUP	EnumRange (OFF ON)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)
priority	CHOICE		EnumRange (NONE PATH-1 PATH-2)
protection	CHOICE		EnumRange (MANUAL AUTO)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (DC voltage 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (DC voltage 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (DC voltage 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Control/Switch board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Voltage control board)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3 (spare))
faults.14	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (COM PSU 1/2)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.486 Novotronik-GTS6303

This page describes the device driver and the device window for a Novotronik LNA Switching Unit GTS6303.

The driver supports the 1:1 and 2:1 redundancy models P/N 1200288 and 1200279. It detects

the number of installed LNAs and sets its parameters/ranges accordingly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch parameters.
- --- Switch readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS6300 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
controlMode	Switches the device control mode between local and remote.
dcPs1	Switches DC power supply 1.
dcPs2	Switches DC power supply 2.
dcPs3	Switches DC power supply 3.
dcBiasMode	Switch DC Bias Control Mode between AUTO and MANUAL.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remote Control

- TCP uses Terminal LF-LFCR protocol
- Default TCP port is 9998

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.controlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.dcPs1	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs2	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs3	CHOICE	SETUP	EnumRange DIS (OFF ON)
config.dcBiasMode	CHOICE	SETUP	EnumRange (MANUAL AUTO)
config.deviceComment	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange DIS (A B)
position.2	CHOICE		EnumRange DIS (A B)
priority	CHOICE		EnumRange DIS (NONE PATH-1 PATH-2)
protection	CHOICE		EnumRange (MANUAL AUTO)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.noLna	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.noSwitches	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (5VDC)
faults.02	ALARM	R/O	AlarmFlagRange R/O (12VDC/1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (12VDC/2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Control/Switch board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (DC supply board Amp 1)
faults.07	ALARM	R/O	AlarmFlagRange R/O (DC supply board Amp 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (DC supply board Amp 3)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch board 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switch board 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch board 3)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Switch board 4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.487 Novotronik-GTS6305

This page describes the device driver and the device window for a Novotronik LNA Power Supply Unit GTS6305.

The driver is intended for monitoring only purpose without it's own parameters page.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS6305 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
config.controlMode	Switches the device control mode between local and remote.
config.dcPs1	Switches DC power supply 1.
config.dcPs2	Switches DC power supply 2.
config.dcPs3	Switches DC power supply 3.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.controlMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.dcPs1	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs2	CHOICE	SETUP	EnumRange (OFF ON)
config.dcPs3	CHOICE	SETUP	EnumRange (OFF ON)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (DC voltage 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (DC voltage 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (DC voltage 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Unit Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Control/Switch board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Voltage control board)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Amplifier 3 (spare))
faults.10	ALARM	R/O	AlarmFlagRange R/O (DC board 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (DC board 2)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DC board 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (COM PSU 1/2)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Module Temperature)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Display)

9.488 Novotronik-GTS6821

This page describes the device driver and the device window for a Novotronik Switching Unit GTS6821

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS6821 do not need any configuration parameters except the settings for the serial interface

parameter	*description*

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position.10	CHOICE		EnumRange (A B)
position.11	CHOICE		EnumRange (A B)
position.12	CHOICE		EnumRange (A B)
position.13	CHOICE		EnumRange (A B)
position.14	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
position.15	CHOICE		EnumRange (A B)
wgstat.10	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.11	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.12	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.13	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.14	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.15	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.10	HEX		HexRange (0 .. 0)
wgcmd.11	HEX		HexRange (0 .. 0)
wgcmd.12	HEX		HexRange (0 .. 0)
wgcmd.13	HEX		HexRange (0 .. 0)
wgcmd.14	HEX		HexRange (0 .. 0)
wgcmd.15	HEX		HexRange (0 .. 0)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch ctrl board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch S10)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Switch S11)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch S12)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch S13)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch S14)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch S15)
initialized	TEXT	R/O	StringRange R/O

9.489 Novotronik-GTS6832

This page describes the device driver and the device window for a Novotronik Switching Unit GTS6832.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTS6832 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position.16	CHOICE		EnumRange (A B)
position.18	CHOICE		EnumRange (A B)
position.24	CHOICE		EnumRange (A B)
position.25	CHOICE		EnumRange (A B)
position.26	CHOICE		EnumRange (A B)
wgstat.16	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.18	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.24	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.25	HEX	R/O	HexRange R/O (0 .. 0)
wgstat.26	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd.16	HEX		HexRange (0 .. 0)
wgcmd.18	HEX		HexRange (0 .. 0)
wgcmd.24	HEX		HexRange (0 .. 0)
wgcmd.25	HEX		HexRange (0 .. 0)
wgcmd.26	HEX		HexRange (0 .. 0)

name	*type*	*flags*	*range*
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal voltage)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch ctrl board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch S16)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Switch S18)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch S24)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch S25)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch S26)
initialized	TEXT	R/O	StringRange R/O

9.490 Novotronik-GTV2455

This page describes the device driver and the device window for a Novotronik Variable Amplifier GTV2455.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Amplifier setting.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The GTV2455 do not need any configuration parameters except the settings for the serial interface. The default address is 0.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
gain	FLOAT		DoubleRange (0 .. 20)
faults.01	ALARM	R/O	AlarmFlagRange R/O (DC Power 1)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (DC Power 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (System)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)

9.491 Novotronik-MAS3160

This page describes the device driver and the device window for a Novotronik 16x16 Transmit Matrix MAS3160.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS3160 do not need any configuration parameters except SNMP settings

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
input.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
input.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
input.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
faults.01	ALARM	R/O	AlarmFlagRange R/O (CRK2000182)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IOK in 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (IOK in 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (IOK in 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IOK in 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HSK 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HSK 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (HSK 3)
faults.10	ALARM	R/O	AlarmFlagRange R/O (HSK 4)
faults.11	ALARM	R/O	AlarmFlagRange R/O (IOK out 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (IOK out 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IOK out 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (IOK out 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2)

9.492 Novotronik-MAS3660

This page describes the device driver and the device window for a Novotronik 64x64 Switch Matrix MAS3660.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS3660 do not need any configuration parameters except SNMP settings

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.17	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.18	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.19	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.20	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.21	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.22	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.23	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.24	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.25	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.26	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.27	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.28	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.29	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.30	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.31	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.32	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.33	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.34	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.35	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.36	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.37	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.38	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.39	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.40	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.41	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.42	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.43	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.44	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.45	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.46	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.47	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.48	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.49	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.50	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.51	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.52	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.53	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.54	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.55	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.56	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.57	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.58	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.59	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.60	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.61	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.62	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.63	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.64	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
iok.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hlk.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.in.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.33.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.34.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.35.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.36.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.37.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.38.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.39.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.40.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.41.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.42.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.43.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.44.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.45.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.46.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.47.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.48.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.49.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.50.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.51.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.52.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.in.53.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.54.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.55.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.56.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.57.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.58.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.59.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.60.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.61.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.62.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.63.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.64.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.out.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
iok.out.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (CRK2000182)
faults.02	ALARM	R/O	AlarmFlagRange R/O (SCK2000187)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.04	ALARM	R/O	AlarmFlagRange R/O (IOK Input cards)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HLK Divider cards)
faults.06	ALARM	R/O	AlarmFlagRange R/O (HSK Matrix cards)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HSK Switch cards)
faults.08	ALARM	R/O	AlarmFlagRange R/O (IOK Output cards)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2A)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1B)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2B)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1A-1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1A-2)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2A-1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2A-2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1B-1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1B-2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2B-1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2B-2)

9.493 Novotronik-MAS3670

This page describes the device driver and the device window for a Novotronik 32x64 Switch Matrix MAS3670.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS3670 do not need any configuration parameters except SNMP settings

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read

community expected by the device.

- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.17	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.18	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.19	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.20	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.21	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.22	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.23	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.24	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.25	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.26	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.27	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.28	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.29	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.30	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
output.31	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)

name	*type*	*flags*	*range*
output.32	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64)
iok.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hlk.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hlk.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.17.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.in.18.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.19.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.20.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.21.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.22.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.23.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.24.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.25.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.26.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.27.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.28.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.29.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.30.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.31.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.in.32.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.09.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.10.fault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
hsk.out.11.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.12.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.13.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.14.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.15.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
hsk.out.16.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.01.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.02.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.03.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.04.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.05.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.06.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.07.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
iok.out.08.fault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (CRK2000182)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IOK Input cards)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HLK Divider cards)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HSK Matrix cards)
faults.06	ALARM	R/O	AlarmFlagRange R/O (HSK Switch cards)
faults.07	ALARM	R/O	AlarmFlagRange R/O (IOK Output cards)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1-1)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1-2)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2-1)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2-2)

9.494 Novotronik-MAS5321

This page describes the device driver and the device window for a Novotronik Matrix MAS5321. It support models with 16, 24 or 32 inputs and 32 outputs:

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS5321 do not need any configuration parameters except the settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.b	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.04	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.05	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.07	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.10	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.16	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.19	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.21	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.22	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.23	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.24	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.25	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.26	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.27	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.28	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.29	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.30	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.31	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.32	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Manual ctrl board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Input card 1/2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Input card 3/4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Input card 5/6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Input card 7/8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Distribution card 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Distribution card 2)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Distribution card 3)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Distribution card 4)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Distribution card 5)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Distribution card 6)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Distribution card 7)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Distribution card 8)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Matrix card 1)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Matrix card 2)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Matrix card 3)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Matrix card 4)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Matrix card 5)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Matrix card 6)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Matrix card 7)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Matrix card 8)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch card 1)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch card 2)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch card 3)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch card 4)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch card 5)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch card 6)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch card 7)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Switch card 8)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Output card 1)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Output card 2)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Output card 3)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Output card 4)
reset	TEXT	NOPRESET	StringRange
internal.manual	TEXT	R/O	StringRange R/O
internal.board	TEXT	R/O	StringRange R/O
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.495 Novotronik-MAS5513

This page describes the device driver and the device window for a Novotronik 32x2 Matrix from type MAS5513.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS5513 do not need any configuration parameters except the settings for the IP address and SNMP settings

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for may be *10.0.0.1 public private* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02)
output.02	CHOICE		EnumRange (00 01 02)
output.03	CHOICE		EnumRange (00 01 02)
output.04	CHOICE		EnumRange (00 01 02)
output.05	CHOICE		EnumRange (00 01 02)
output.06	CHOICE		EnumRange (00 01 02)
output.07	CHOICE		EnumRange (00 01 02)
output.08	CHOICE		EnumRange (00 01 02)
output.09	CHOICE		EnumRange (00 01 02)
output.10	CHOICE		EnumRange (00 01 02)
output.11	CHOICE		EnumRange (00 01 02)
output.12	CHOICE		EnumRange (00 01 02)
output.13	CHOICE		EnumRange (00 01 02)
output.14	CHOICE		EnumRange (00 01 02)
output.15	CHOICE		EnumRange (00 01 02)
output.16	CHOICE		EnumRange (00 01 02)
output.17	CHOICE		EnumRange (00 01 02)
output.18	CHOICE		EnumRange (00 01 02)
output.19	CHOICE		EnumRange (00 01 02)
output.20	CHOICE		EnumRange (00 01 02)
output.21	CHOICE		EnumRange (00 01 02)
output.22	CHOICE		EnumRange (00 01 02)
output.23	CHOICE		EnumRange (00 01 02)

name	*type*	*flags*	*range*
output.24	CHOICE		EnumRange (00 01 02)
output.25	CHOICE		EnumRange (00 01 02)
output.26	CHOICE		EnumRange (00 01 02)
output.27	CHOICE		EnumRange (00 01 02)
output.28	CHOICE		EnumRange (00 01 02)
output.29	CHOICE		EnumRange (00 01 02)
output.30	CHOICE		EnumRange (00 01 02)
output.31	CHOICE		EnumRange (00 01 02)
output.32	CHOICE		EnumRange (00 01 02)
faults.01	ALARM	R/O	AlarmFlagRange R/O (CRK 2000286)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Manual Control)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IOK rx-in 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HLK rx 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HLK rx 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (HLK rx 3)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HLK rx 4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 3)
faults.11	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 4)
faults.12	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 5)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 6)
faults.14	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 7)
faults.15	ALARM	R/O	AlarmFlagRange R/O (IOK rx-out 8)
faults.16	ALARM	R/O	AlarmFlagRange R/O (IOK tx-out 1)
faults.17	ALARM	R/O	AlarmFlagRange R/O (HLK tx 1)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (HLK tx 2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (HLK tx 3)
faults.20	ALARM	R/O	AlarmFlagRange R/O (HLK tx 4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (IOK tx-in 8)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Power supply +9V/2)

9.496 Novotronik-MAS5514

This page describes the device driver and the device window for a Novotronik Matrix MAS5514 . The RX matrix section switches and distributes RF signals from 12 sources (inputs) to 8 receivers (outputs), the TX matrix section switches and combines RF signals from 8 sources (inputs) to 8 receivers (outputs).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Port switching.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The MAS5321 do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,8,1* .

Remote Control

- RS485/422 uses Novotronik-ASCII protocol
- RS232 uses Novotronik protocol
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)

name	*type*	*flags*	*range*
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08)
input.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
input.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12)
info.swVersion	TEXT	R/O	StringRange R/O
info.swDate	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.unitInfo	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Internal communication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Red controller board)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 1 +5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply 2 +5V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply 1 +9V)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power supply 2 +9V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Manual ctrl module)
faults.08	ALARM	R/O	AlarmFlagRange R/O (RX input 1..4)
faults.09	ALARM	R/O	AlarmFlagRange R/O (RX input 5..8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RX input 9..12)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RX 8x8 board 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RX 8x8 board 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RX output 1..4)
faults.14	ALARM	R/O	AlarmFlagRange R/O (RX output 5..8)
faults.15	ALARM	R/O	AlarmFlagRange R/O (TX input 1..4)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TX input 5..8)
faults.17	ALARM	R/O	AlarmFlagRange R/O (TX 8x8 board 1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (TX output 1..4)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TX output 5..8)
reset	TEXT	NOPRESET	StringRange

9.497 Novotronik-Matrix24

This page describes the device driver and the device window for a Novotronik L-Band Matrix with up to 24 inputs and outputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Novotronik Matrix do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)

name	*type*	*flags*	*range*
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.17	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.18	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.19	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.20	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.21	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)

name	*type*	*flags*	*range*
output.22	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.23	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
output.24	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Batterie status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply red.)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V/1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V/2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Manual ctrl board)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Interface board)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch board 01)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch board 02)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch board 03)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Switch board 04)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Switch board 05)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switch board 06)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch board 07)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switch board 08)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch board 09)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Switch board 10)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Switch board 11)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Switch board 12)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch board 13)

name	*type*	*flags*	*range*
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch board 14)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch board 15)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch board 16)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch board 17)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch board 18)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch board 19)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch board 20)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch board 21)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch board 22)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch board 23)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch board 24)
reset	TEXT		StringRange
internal.manual	TEXT	R/O	StringRange R/O
internal.board	TEXT	R/O	StringRange R/O
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.498 Novotronik-Matrix32

This page describes the device driver and the device window for a Novotronik L-Band Matrix with up to 32 inputs and outputs. The driver reads the available inputs (in steps of 8) and outputs (in steps of 1) from the matrix and shows only these available ports in the user interface.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Novotronik Matrix do not need any configuration parameters except the settings for the serial interface

parameter	***description***
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.02	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.03	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.04	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.05	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.06	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.07	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.08	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.09	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.10	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.11	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.12	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.13	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.14	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.15	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.16	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.17	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.18	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.19	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.20	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.21	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.22	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.23	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.24	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.25	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.26	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.27	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)

name	*type*	*flags*	*range*
output.28	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.29	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.30	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.31	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
output.32	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Batterie status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply red.)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply +5V/2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V/1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power supply +12V/2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Manual ctrl board)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Interface board)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch board 01)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch board 02)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch board 03)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Switch board 04)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Switch board 05)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Switch board 06)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch board 07)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switch board 08)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch board 09)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Switch board 10)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Switch board 11)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Switch board 12)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch board 13)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch board 14)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch board 15)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch board 16)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch board 17)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch board 18)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Switch board 19)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Switch board 20)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Switch board 21)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Switch board 22)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Switch board 23)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Switch board 24)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Switch board 25)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Switch board 26)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Switch board 27)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Switch board 28)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Switch board 29)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Switch board 30)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Switch board 31)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Switch board 32)
reset	TEXT	NOPRESET	StringRange
internal.manual	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.board	TEXT	R/O	StringRange R/O
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.499 Novotronik-Matrix8

This page describes the device driver and the device window for a Novotronik L-Band Matrix with up to 8 inputs and outputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Novotronik Matrix do not need any configuration parameters except the settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swversion	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Battery status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Manual ctrl board)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch board 01)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch board 02)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch board 03)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch board 04)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Switch board 05)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switch board 06)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Switch board 07)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch board 08)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
internal.manual	TEXT	R/O	StringRange R/O
internal.board	TEXT	R/O	StringRange R/O
internal.modules	TEXT	R/O	StringRange R/O
internal.power	TEXT	R/O	StringRange R/O

9.500 Novus-ND2310D

This page describes the device driver and the device window for the Novus ND2310D 10-Channel Distribution Amplifier via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Device settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- nsCommandDelay --- Defines the delay the driver waits after querying a parameter using the nsCommand OID. This is used for position, protection, input and fault thresholds as well as channel reference values.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.identifier	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.nsCommandDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
inputAVrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inputBVrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
location	CHOICE	R/O	EnumRange R/O (1 2)
channel1Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel2Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel3Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel4Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel5Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel6Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel7Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel8Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel9Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel10Vrms	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuDigPos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuAnaPos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuAnaNeg	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inputAThreshold	FLOAT		DoubleRange (0.05 .. 1.00)
inputBThreshold	FLOAT		DoubleRange (0.05 .. 1.00)
faultAThreshold	FLOAT		DoubleRange (0.05 .. 0.95)
faultBThreshold	FLOAT		DoubleRange (0.05 .. 0.95)
channel1Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel2Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel3Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel4Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel5Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel6Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel7Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel8Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel9Ref	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
channel10Ref	FLOAT		DoubleRange (0.00 .. 0.00)
latchAvg	TEXT	NOPRESET	StringRange
internal.inp	HEX	R/O	HexRange R/O (0 .. 0)
internal.csw	HEX	R/O	HexRange R/O (0 .. 0)
internal.ps1	HEX	R/O	HexRange R/O (0 .. 0)
internal.ps2	HEX	R/O	HexRange R/O (0 .. 0)
internal.aps	HEX	R/O	HexRange R/O (0 .. 0)
internal.cfb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pc1	HEX	R/O	HexRange R/O (0 .. 0)
internal.pc2	HEX	R/O	HexRange R/O (0 .. 0)
int.position	TEXT	R/O	StringRange R/O
internal.response	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1 -8V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU1 +8V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU1 5V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU1 Communication)
faults.07	ALARM	R/O	AlarmFlagRange R/O (PSU1 DC Power)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PSU1 AC Power)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PSU2 -8V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PSU2 +8V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU2 5V)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU2 Communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU2 DC Power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU2 AC Power)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Potentiometer)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PCB Assembly)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Channel 1 General)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Channel 2 General)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Channel 3 General)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Channel 4 General)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Channel 5 General)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Channel 6 General)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Channel 7 General)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Channel 8 General)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Channel 9 General)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Channel 10 General)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Channel 1 External)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (Channel 2 External)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Channel 3 External)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Channel 4 External)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Channel 5 External)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Channel 6 External)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Channel 7 External)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Channel 8 External)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Channel 9 External)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Channel 10 External)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 1)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 2)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 3)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 4)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 5)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 6)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 7)

name	*type*	*flags*	*range*
faults.68	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 8)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 9)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Primary Amp Ch. 10)
faults.81	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 1)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 2)
faults.83	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 3)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 4)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 5)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 6)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 7)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 8)
faults.89	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 9)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Backup Amp Ch. 10)

9.501 Novus-NR2110-SNMP

This page describes the device driver and the device window for the Novus NR2110-OG Single Channel GPS Locked OCXO Frequency Reference via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Device settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter --- description

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- nsCommandDelay --- Defines the delay the driver waits after querying a parameter using the nsCommand OID. This is used for fault thresholds and channel reference values.
- antennas --- Defines the number of GPS antennas connected (1..2).
- channels --- Defines the number of output channels (1..8).
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.nsCommandDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.channels	INTEGER	SAVE SETUP	IntegerRange (1 .. 8)
config.antennas	INTEGER	SAVE SETUP	IntegerRange (1 .. 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
gps1Lock	BOOLEAN	R/O	BooleanRange R/O (true false)
gps2Lock	BOOLEAN	R/O	BooleanRange R/O (true false)
gps1Satellites	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gps2Satellites	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel1Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel2Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel3Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel4Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel5Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel6Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
channel7Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel8Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu5Vpos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu5Vneg	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuAnt	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuDac	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
faultThreshold	FLOAT		DoubleRange (0.05 .. 0.95)
channel1Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel2Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel3Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel4Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel5Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel6Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel7Ref	FLOAT		DoubleRange (0.00 .. 0.00)
channel8Ref	FLOAT		DoubleRange (0.00 .. 0.00)
latchAvg	TEXT	NOPRESET	StringRange
internal.cflts	HEX	R/O	HexRange R/O (0 .. 0)
internal.pflts	HEX	R/O	HexRange R/O (0 .. 0)
internal.response	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Channel 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Channel 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Channel 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Channel 6)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Channel 7)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Channel 8)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Antenna A)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Antenna B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU -5V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU +5V)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU OCXO)
faults.15	ALARM	R/O	AlarmFlagRange R/O (GPS1 Lock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (GPS2 Lock)

9.502 Novus-NR2110

This page describes the device driver and the device window for the Novus NR2110-OG Single Channel GPS Locked OCXO Frequency Reference via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Device settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
channels	Defines the number of output channels (1..8).
antennas	Defines the number of GPS antennas connected (1..2).

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.channels	INTEGER	SAVE SETUP	IntegerRange (1 .. 8)
config.antennas	INTEGER	SAVE SETUP	IntegerRange (1 .. 2)
gps1Lock	BOOLEAN	R/O	BooleanRange R/O (true false)
gps2Lock	BOOLEAN	R/O	BooleanRange R/O (true false)
gps1Satellites	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gps2Satellites	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel1Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel2Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel3Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel4Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel5Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel6Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel7Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
channel8Vpp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu5Vpos	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psu5Vneg	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
psuOCXO	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faultThreshold	FLOAT		DoubleRange (0.05 .. 0.95)
latchAvg	TEXT	NOPRESET	StringRange
internal.cflts	HEX	R/O	HexRange R/O (0 .. 0)
internal.pflts	HEX	R/O	HexRange R/O (0 .. 0)
internal.dummy	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Channel 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Channel 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Channel 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Channel 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Channel 6)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Channel 7)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Channel 8)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Antenna A)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Antenna B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU -5V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU +5V)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU OCXO)

9.503 Number-Select

The Number-Select logical device acts like a the chip-select of IC's or some kind of address selection. This mean it sets the selected output to true and all other outputs to false.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The position selection page.

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

There is no configuration necessary.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.output.01	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.02	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.03	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.04	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.05	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.06	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.07	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.08	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.09	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.10	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.11	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.12	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
info.output.13	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.14	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.15	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.16	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.17	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.18	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.19	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.20	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.21	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.22	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.23	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.24	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.25	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.26	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.27	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.28	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.29	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.30	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.31	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.32	BOOLEAN	R/O	BooleanRange R/O (true false)
position	INTEGER	SAVE	IntegerRange (0 .. 32)

9.504 Number-Set

This page describes the logical Number Set device driver. It includes 32 output variables. The driver toggles the selected output between true and false and leaves all other outputs in the last state.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch position.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.output.01	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.02	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.03	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.04	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.05	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.06	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.07	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.08	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.09	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.10	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.11	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.12	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
info.output.13	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.14	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.15	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.16	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.17	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.18	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.19	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.20	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.21	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.22	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.23	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.24	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.25	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.26	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.27	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.28	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.29	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.30	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.31	BOOLEAN	R/O	BooleanRange R/O (true false)
info.output.32	BOOLEAN	R/O	BooleanRange R/O (true false)
position	INTEGER		IntegerRange (0 .. 32)

9.505 ORB-Step-Track

The ORB-Step-Track device driver is a customized version of the driver for the ND SatCom Steptrack System. There is only one difference of this driver to the original one: It sets a fault if the step track unit falls back to STANDBY for some reason.

For further information about this device driver see the help page of theNDSatCom-Step-Track

device driver.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.c	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.d	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.e	INTEGER	R/O	IntegerRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0 .. 0)
trackMode	CHOICE		EnumRange (STANDBY REMOTE POSITION MODEL TRACK (NORM) MODEL TRACK (LEARN) MODEL TRACK (PEAKING) SCAN BEACON)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU outdoor unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PL limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Volt meter)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Remote/peaking)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Switched to standby)

9.506 Off-Switch

This logical device send an OFF command everytime a parameter changes.

Typical application is to react on any kind of switch (e.g. waveguide switch or matrix port) and switch transmit equipment OFF (e.g. HPA or Modulator).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	*description*
inputId	Enter the message identifier here of the parameter the device shall listen
offId	Enter the message identifier here of the parameter which should receive the OFF commmand

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputId	TEXT	SAVE SETUP	StringRange
config.offId	TEXT	SAVE SETUP	StringRange
internal.firstRun	TEXT	R/O	StringRange R/O

9.507 OnTarget

This logical device compares up to three target values agains actual values to check if they are within the defined range.

Typical applications are to check if an antenna is moved to the desired target (azimuth, elevation and polarization) or to check if a system reached a commanded temperature.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	*description*
state-A.Id	Enter the message identifier here of the parameter to get the first actual value
state-B.Id	Enter the message identifier here of the parameter to get the second actual value
state-C.Id	Enter the message identifier here of the parameter to get the third actual value

parameter	*description*
target-A.Id	Enter the message identifier here of the parameter which contains the first target value
target-B.Id	Enter the message identifier here of the parameter which contains the second target value
target-C.Id	Enter the message identifier here of the parameter which contains the third target value
delta-A	Defines the range where the first value is within the target range
delta-B	Defines the range where the first value is within the target range
delta-C	Defines the range where the first value is within the target range

remarks

Even if you want to check only 1 or 2 target values you have to configure all 3 message identifiers to get the desired summary output. To achieve this you could simply duplicate the message identifiers and set the delta value to 0.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.state-A.Id	TEXT	SAVE SETUP	StringRange
config.state-B.Id	TEXT	SAVE SETUP	StringRange
config.state-C.Id	TEXT	SAVE SETUP	StringRange
config.target-A.Id	TEXT	SAVE SETUP	StringRange
config.target-B.Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.target-C.Id	TEXT	SAVE SETUP	StringRange
config.delta-A	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
config.delta-B	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
config.delta-C	FLOAT	SAVE SETUP	DoubleRange (0.00000 .. 0.00000)
state.a	FLOAT		DoubleRange (0.00000 .. 0.00000)
state.b	FLOAT		DoubleRange (0.00000 .. 0.00000)
state.c	FLOAT		DoubleRange (0.00000 .. 0.00000)
target.a	FLOAT		DoubleRange (0.00000 .. 0.00000)
target.b	FLOAT		DoubleRange (0.00000 .. 0.00000)
target.c	FLOAT		DoubleRange (0.00000 .. 0.00000)
onTarget	BOOLEAN	R/O	BooleanRange R/O (true false)
onTarget.a	BOOLEAN	R/O	BooleanRange R/O (true false)
onTarget.b	BOOLEAN	R/O	BooleanRange R/O (true false)
onTarget.c	BOOLEAN	R/O	BooleanRange R/O (true false)

9.508 OpenGear-Frame

This page describes the device driver and the device window for an open gear frame The driver is a simple monitoring only driver reading basic information (e.g. vendor, software release, serial number etc.)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and 'private' as write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for an open gear frame may be *10.0.0.1 public private none* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.supplier	TEXT	R/O	StringRange R/O
info.product	TEXT	R/O	StringRange R/O
info.software	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.serialboard	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.firmware	TEXT	R/O	StringRange R/O
info.temp.ambient	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temp.psu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temp.psu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan.1.psu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan.1.psu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan.2.psu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.fan.2.psu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.current.psu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.current.psu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.status	HEX	R/O	HexRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Hardware status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.509 Output

This page describes the device driver and the device window for the*Output* logical device. This device controls one single output port of a PLC device and optionally monitors one or two position indication contacts.

The *Output* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Sets / shows the (translated) PLC port state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
drivePort	The address of the PLC port to set. See the paragraph 'Port addressing' below for details.
indication0Port	The address of the 'position 0' indication port to monitor. If this port address is defined, the device checks if this port is set ('1') after the drivePort has been commanded '0'. See below for more information position indication handling of the device.
indication1Port	The address of the 'position 1' indication port to monitor. If this port address is defined, the device checks if this port is set ('1') after the drivePort has been commanded '1'.
positions	By default the device reports the state of the output as '0'/'1'. You may specify arbitrary texts for these states. 'positions' contains the two descriptions, separated by a comma. If you customize this setting, you should consider that other logical devices (e.g. a protection switch) may set standard values like A, B, OFF or ON when commanding this value.
commFaultId	By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault.

parameter	***description***
The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).	

Fault flags

fault flag	***fault name***	***description***
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Position Indication	This fault is set if at least one of configured position indication inputs reads a state which does not match commanded switch position.
faults.02	Configuration Setup	This fault signals that the 'drivePort' parameter ist not properly configured. The device expects, that 'drivePort' is set to a value which matches the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the**+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address ' *FANUC.Q00065+1* ' refers to the *2^1* bit in the variable ' *Q00065* '

provided by the ' *FANUC* ' device. In fact this addresses the output table register '%Q00066' at the PLC. Other PLC drivers may use a different naming scheme for the variables, but the concept always is the same.

Position indication inputs

The *Output* device optionally may be configured to use position indication inputs to verify the commanded position of a (RF-) switch/relay connected to the output. The device monitors these inputs in the following way:

- If the port address for an indication input is left blanc, the state of this port is assumed to be OK.
- Indication inputs may be used with negative logic, simply swap the ports in this case.
- The *Output* device always reports the output position as it was commanded, not a position reported by an indication input.
- Monitoring and checking the position indication inputs happens all the time, but after a new output position has been commanded this is paused for 1..2 seconds to give the switch time to reach it's final position.
- The actual state of the position indication inputs is displayed at the 'info' page of the device window.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.indication	TEXT	R/O	StringRange R/O
config.drivePort	TEXT	SAVE SETUP	StringRange
config.indication0Port	TEXT	SAVE SETUP	StringRange
config.indication1Port	TEXT	SAVE SETUP	StringRange
config.positions	TEXT	SAVE SETUP	StringRange
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (0 1)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Position Indication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.510 Paradise-RCP2-1100-SNMP

This page describes the device driver and the device window for the Paradise RCP2 1100 redundancy controller. Compared to the Paradise-RCP2-SNMP it includes the SSPA control and monitoring.

It was tested with the following configurations:

- RCP2-1100 with Firmware V.616,
  - 2 SSPA, 1 WG Switch, 1+1 redundancy, no LNA/LNB
- RCP2-1100 with Firmware V.616,
  - 1 WG Switch only

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch and redundancy parameters.
- --- The HPA parameters if supported.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port

number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- hasHPA --- Select if HPAs could be directly monitored via RCP.
- rfUnits --- Select the rf units between dBm or Watt.
- inhibit --- Select the inhibit usage of the unit (OFF/INTERNAL/EXTERNAL/ALL).

Communication protocol / Interface type

This device driver requires the SNMP protocol. Please remember to configure the ip settings and enable SNMP at the front panel

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.deviceId	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.deviceType	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.s1Position	TEXT	R/O	StringRange R/O
info.hpaType	CHOICE		EnumRange (OFF CO RM PATH VBUC SYSX PMAX)
info.sysType	CHOICE		EnumRange (1:2 REDUNDANT 1:1 REDUNDANT 1:1 PHASE COMBINED DUAL 1:1 SINGLE SWITCH 1:2 PHASE COMBINED)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.hasHPA	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.rfUnits	CHOICE	SETUP	EnumRange (dBm W)
config.inhibit	CHOICE	SETUP	EnumRange (OFF INTERNAL EXTERNAL ALL)
position	CHOICE		EnumRange (A B)
location	CHOICE		EnumRange (1 2)
protection	CHOICE	SAVE	EnumRange (AUTO MANUAL)
meas.ps1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.ps2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE		EnumRange DIS (OFF ON)
tx.attn	FLOAT		DoubleRange DIS (0.0 .. 20.0)
tx.1.attn	FLOAT		DoubleRange DIS (0.0 .. 20.0)
tx.2.attn	FLOAT		DoubleRange DIS (0.0 .. 20.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
tx.1.measuredLevel	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
tx.2.measuredLevel	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.1.fwdPwr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
meas.2.fwdPwr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
meas.1.temp	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.2.temp	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.1.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.2.current	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.ambientTemp	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
internal.s1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Switch 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Model not supported)
internal.lo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.lorf	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.hi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

9.511 Paradise-RCP2-SNMP

This page describes the device driver and the device window for the Paradise RCP2 redundancy controller. It was tested as a simple switch controller only.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller

parameter	***description***
address	The device IP address and the SNMP community strings as set at the device's front panel.

Communication protocol / Interface type

This device driver requires the SNMP protocol. Please remember to configure the ip settings and enable SNMP at the front panel

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.deviceType	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.s1Position	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position	CHOICE		EnumRange (A B)
location	CHOICE		EnumRange (1 2)
protection	CHOICE	SAVE	EnumRange (AUTO MANUAL)
internal.s1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RF Switch 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)

9.512 Paradise-RCP2

This page describes the device driver and the device window for the Paradise RCP2 redundancy controller with serial interface Tested with a RCP2 unit which handles online one WG-Switch and no redundancy

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the unit

parameter	*description*
address	Address of the unit

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.deviceType	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange (MANUAL AUTO)
priority	CHOICE		EnumRange (POL1 POL2)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Auxiliary)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB3)

9.513 Paradise-SSPA-SNMP

This page describes the device driver and the device window for the Paradise C-Band indoor SSPA amplifier. It was tested in a 1+1 redundancy configuration.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational upconverter parameters
- --- Meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the SSPA.

parameter	*description*
address	The device IP address and the SNMP community strings as set at the device's front panel.

parameter	*description*
powerDisplayMode	Select if you want to see gain or attenuation
rfUnits	Switch the display mode between dBm and W. This changes the front panel display and the messages of the remote control protocol, but the drivers display both at any time.
sysMode	Allows to select the system configuration.
standbyMode	Allows to select the standby mode in a redundant configuration.

Communication protocol / Interface type

This device driver requires the SNMP protocol. Please remember to configure the ip settings (Main Menu - 2.Com Setup) and enable SNMP at the front panel

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.deviceType	TEXT	R/O	StringRange R/O
info.ioBoardRev	TEXT	R/O	StringRange R/O
info.digitalCoreRev	TEXT	R/O	StringRange R/O
info.s1Position	TEXT	R/O	StringRange R/O

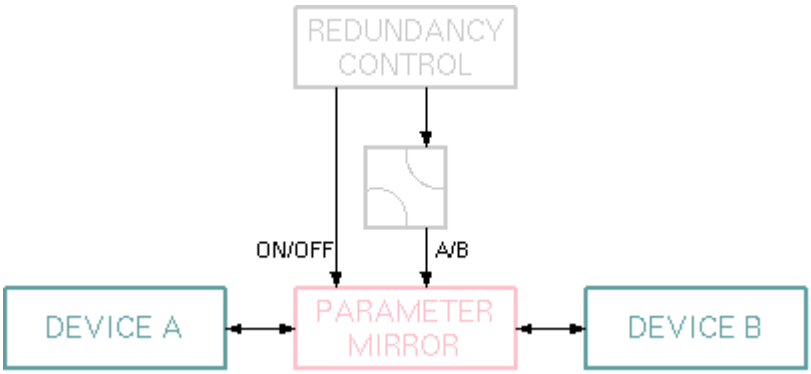
name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.rfUnits	CHOICE	SETUP	EnumRange (dBm W)
config.sysMode	CHOICE	SETUP	EnumRange (StandAlone 1:1 1:2)
config.standbyMode	CHOICE	SETUP	EnumRange (HotStandby ColdStandby)
config.powerSupplies	CHOICE	SAVE SETUP	EnumRange (1 2)
config.hasRefPwrSensor	CHOICE	SAVE SETUP	EnumRange (YES NO)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.ps1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.ps2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.booster1Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.booster2Voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
red.mode	CHOICE		EnumRange DIS (AUTO MANUAL)
red.switch	CHOICE		EnumRange DIS (STANDBY ONLINE)
internal.fwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.ref	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.s1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High temp)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low regulator voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low DC current)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Auxiliary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (BUC)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Module 1)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Module 2)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Module 3)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Module 4)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Cooling fan)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Low forward RF)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High reflected RF)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch position 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Switch position 2)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.514 Parameter-Mirror

The Parameter-Mirror logical device ties a single parameter of two 1:1 redundant devices together. Any time the parameter is changed at the primary device, the Parameter-Mirror

copies this setting to the spare device. The Parameter-Mirror itself provides no redundancy control or switching, it is intended to be used as an extension of an existing redundancy switching unit.



The Parameter-Mirror works symmetrically, there is no fixed definition of a primary and a spare device. To determine it's working direction, the Parameter-Mirror monitors a switch position. This switch position may be an arbitrary M&C parameter, hence a hardware switch may be monitored as well as the output state of a software based redundancy control. Accordingly, the redundancy controller may be both, a sat-nms logical device or an external hardware unit.

Mirroring the parameter usually only is desired while the redundancy / protection switch is activated. The Parameter-Mirror monitors a second parameter, which enables mirroring. This usually is the ON/OFF setting of the redundancy controller. While the Parameter-Mirror is disabled by this parameter, no mirroring of settings takes place. However, the Parameter-Mirror monitors the parameter setting at both devices and remembers them in an internal memory. Once the redundancy switch is enabled, the Parameter-Mirror copies the setting of the primary device to the spare one.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Parameter-Mirror device with the devices it shall mirror.

parameter	***description***
enableId	Enter the message identifier here of the parameter the Parameter-Mirror shall monitor in order to enable / disable the mirroring function. (e.g. "PROTECT.enable")

parameter	*description*
enableVal	This is the message contents, the Parameter-Mirror expects to enable the mirroring. (e.g. "ON")
directionId	Enter the message identifier here of the switch position which determines the direction of the mirroring function. (e.g. "S2.position")
direction1To2Val	This is the message contents, the Parameter-Mirror expects to mirror from device 1 to device 2. (e.g. "A")
parameter1Id	The message identifier of the parameter 1 to mirror.
parameter2I	The message identifier of the parameter 1 to mirror.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.enableVal	TEXT	R/O	StringRange R/O
info.direction1To2Val	TEXT	R/O	StringRange R/O
info.forward1to2	BOOLEAN	R/O	BooleanRange R/O (true false)
info.forward2to1	BOOLEAN	R/O	BooleanRange R/O (true false)
config.enableId	TEXT	SAVE SETUP	StringRange
config.enableVal	TEXT	SAVE SETUP	StringRange
config.directionId	TEXT	SAVE SETUP	StringRange
config.direction1To2Val	TEXT	SAVE SETUP	StringRange
config.parameter1Id	TEXT	SAVE SETUP	StringRange
config.parameter2Id	TEXT	SAVE SETUP	StringRange

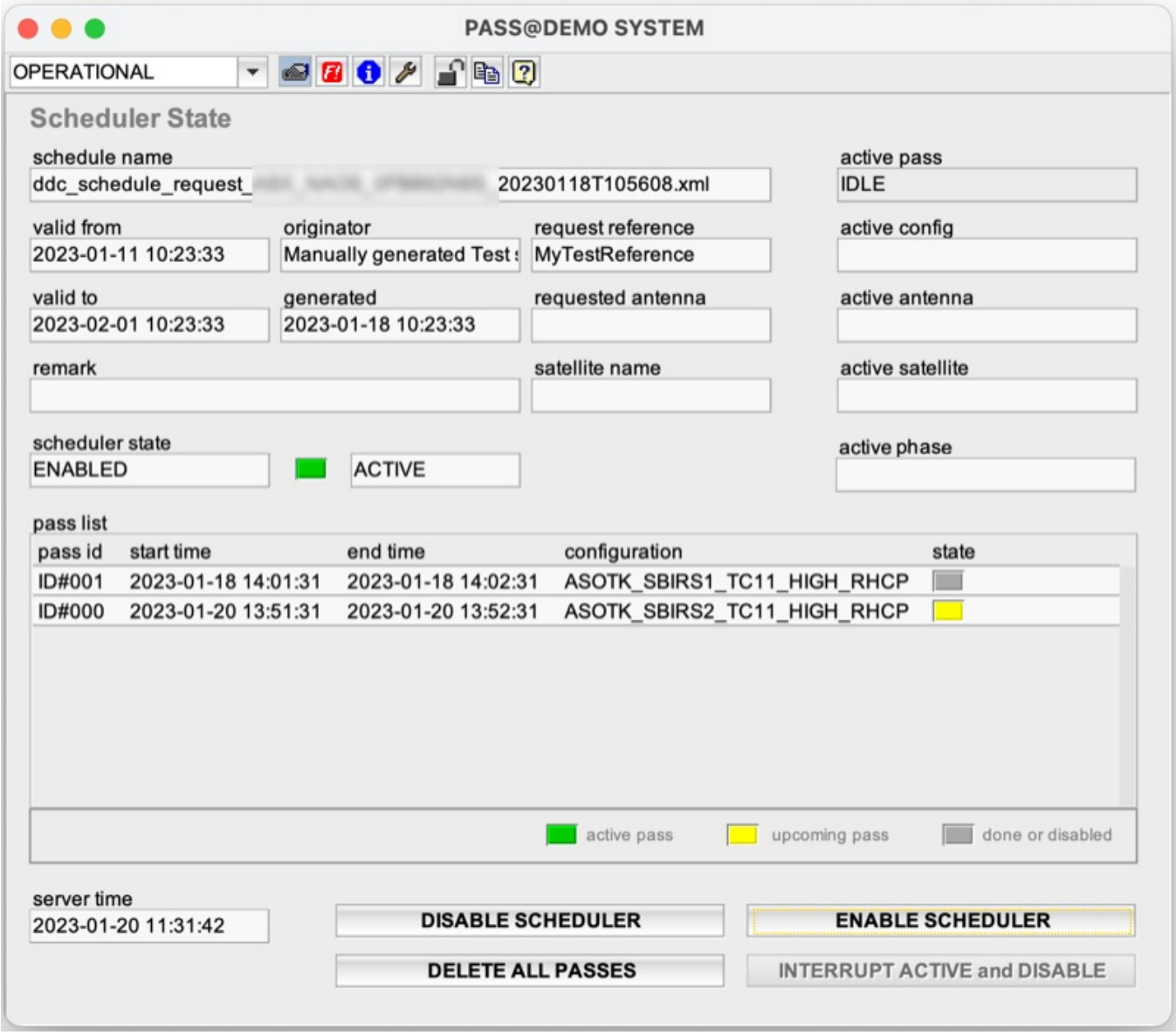
9.515 Pass-Scheduler

The Pass-Scheduler logical device provides a very specialized version of sat-nms system scheduler. The Pass-Scheduler gets its information from an XML file with times when an

overflying satellite rises and sets. For each pass pf the satellite the scheduler generates four events (PRE-PASS, START-PASS, END-PASS and POST-PASS) at times which correlate with the satellite pass times way globally configured for the device in a config file. Each event triggers a macro with a special name, the name is generated from parameters read from the XML schedule definition

The device scans a configurable directory for XML schedule files and uses the first one found as the new schedule. Then the XML schedule file gets moved to an archive directory to prevent the device from reading it again.

The scheduler embedded in this device is always active, neglecting any OUT-OF-SERVICE setting of the device. Setting the device OUT-OF-SERVICE only prevents it from scanning the schedule directory, this does not disable the scheduler itself. The recommended way to disable the scheduler is to use the "Disable" / "Interrupt" buttons in the device window.



The Pass-Scheduler principally requires no interaction with the operator. Once set up the Pass-Scheduler does its work in background, The device window of the Pass-Scheduler nevertheless provides some state information and controls:

- The schedule display shows the parameters of the actually loaded schedule and lists the passes contained in this schedule together with their individual state.
- **DELETE** --- Deletes all passes from the schedule except the active pass. This is the same as loading an empty schedule.
- **DISABLE** --- Disables all future passes in the schedule but not the active pass. Disabled passes remain in the schedule, but no events are executed for these passes the their due time.
- **INTERRUPT** --- Interrupts the schedule, forces the active pass to become disabled.
- **ENABLE** --- This button enables all disabled passes. Passes which became outdates while disabled directly become "done". The ENABLE button never enables a pass which immediately would turn to the active pass.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the actual schedule state and provides buttons to delete / interrupt / disable or enable the actual schedule
- --- The fault state of the device.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
antennaId	The XML schedule file may contain satellite pass information to be interpreted by multiple Antennas. Hence each schedule request in the file is marked for which antenna the request applies. antennaId must be set to then ID if the antenna this scheduler shall work for. Note, this is not necessarily the sat-nms device name of the antenna, but the antenna ID used by the originator of the XML schedule file. If this parameter is not set, the device will ignore all pass definition in the file
scheduleDir	The name of the directory, the device shall scan for new XML schedule files. The name may be an absolute directory path or one relative to the working directory of the satnms software. The directory must exist and must be permanently reachable for the software.

parameter	*description*
archiveDir	The name of a directory the device shall move an XML schedule file to after it has been read. Like scheduleDir, the directory path may be absolute or relative. The archive directory must exist and it must be permanently writable for the software. It must refer to the same physical medium as the scheduleDir. scheduleDir and archiveDir must not be the same or the device will read XML schedule files over and over again.

Pass Phases

The Pass-Scheduler device manages four phases of a satellite pass: PRE, START, END and POST. each phase starts a defined time relative to the calculated start / end times of the satellite pass in the schedule. With the beginning of each phase a macro is called which executes the actions to be done in this phase

phase	*displayed name*	*description*
PRE	prepass	prepares the equipment for the upcoming pass
START	startpass	starts the pass, e.g. switches RF-ON
END	endpass	ends the pass, e.g. switches RF-OFF
POST	postpass	cleans up, returns the equipment to idle state

The *state.phase* parameter shows the actually running phase while a pass is active with the abbreviations shown in column *displayed name* in the table above. Please note that the *postpass* phase often may not be visible as the scheduler returns to*idle* immediately after it entered *postpass*.

Configuration file

The device reads a configuration file at startup which defines prefixes for the macro names to be used for the four events of a satellite pass. This file also defines the time offsets of the schedules events to the pas times of the satellite.

The configuration file has the same name as the device with the file extension ".cfg". the example below illustrates the the structure if the file:

```
# configures the pass scheduler macro name prefixes and relative timings
# there may be comment lines starting with '#' at the top of the file, the
# definitions of the pass phases however must appear without interruption
#
000_SCHD_prepass_, StartTime, -900
000_SCHD_startpass_, StartTime, 0
000_SCHD_endpass_, EndTime, 0
000_SCHD_postpass_, EndTime, +300
```

The file consists of four lines defining the parameters for the PRE-PASS, START-PASS, END-PASS and POST-PASS events. Each line consists of three definitions separated by comma characters. The first is the name prefix used for the macros related to this event, the second is one of the keywords StartTime or EndTime, denoting the reference from where the time of this event shall be calculated. The last definition in the line is the offset of the time when the event is scheduled, relative to the reference in the second parameter.

The file may have comment lines starting with a '#' character at its top, but this is optional. There mus not be any comments between the four lines defining the event parameters.

Macro names

The names of the macros executed by the scheduler for the four events of each pass are built from the prefix defined for the particular event in the configuration file described above and the actual "configuration" for this pass as shown in the pass list in the column "configuration". This configuration string is built from five parameters read from the XML definition of this pass. Example

```
000_SCHD_startpass_SX_YES_TC1_HIGH_RHCP
```

is the name of the macro to be executed with the START-PASS event for a pass defined with the parameters bands=SX, uplink=YES, bb-id=TC1, dl-rate=HIGH and polarization=RHCP (assuming the macro name prefix as used in the example above).

Persistent schedule

The device stores its schedule in a JSON formatted file with the name of the device and the file extension ".json". The file is read when the software starts, the schedule persists over stop / start cycles of the software this way.

XML schedule replacement

The device scans a configurable directory for XML schedule files and uses the first one found as the new schedule. Then the XML schedule file gets moved to an archive directory to prevent the device from reading it again.

When a new schedule is read, it replaces the existing schedule. If from the old schedule there is an active pass in progress when the new schedule is read, this pass will be completed. If the new schedule contains a pass which collides in some way with the active pass, the schedule

will not be read and a fault will be raised.

XML schedule file name

When the device scans the schedule directory, it looks for files looking like

ddc_schedule_request_<recipient><*satellite*><request_ref>_<generation_time>.xml

regex: 'ddc_schedule_request_[A-z]+*[A-z0-9]+*[A-z0-9]+_[0-9] {8} T[0-9] {6} .xml'

Files not matching this pattern are ignored.

Fault flags

The device knows a number of fault flags which are set if some errors occur when reading a schedule file. The list below gives som explanations regarding these fault flags:

- **faults.01** --- (Events missed) The M&C software missed to execute some schedule events while if was not running. This fault may be detected after starting the M&C software to signal that the equipment
- **faults.02** --- (Events dropped) The M&C software started with a partially missed pass. In this case the remaining events for this pass (e.g. "end" and "post") have been dropped.
- **faults.03** --- (Schedule load) There was a general problem reading the schedule file. This typically is a parsing problem e.g. if the schedule file has been truncated and no longer is a valid XML document.
- **faults.04** --- (Pass outside) The schedule contains passes with time that are outside the validity period of the schedule (as stated in tis header). These passes have been ignored.
- **faults.05** --- (Swapped pass times) There are passes in the schedule which state end end time before the start time. These passes have been ignored.
- **faults.06** --- (Overlapping passes) Loading the schedule would lead to overlapping passes, either between two passes of the schedule to load or between the currently active pass and a pass of the schedule to load. The schedule is not loaded in this case, the previous schedule remains unchanged.
- **faults.07** --- (Overlaps current time) The schedule to load contains a pass which immediately would become the active pass with at least the "pre" event missing. The schedule is not loaded in this case, the previous schedule remains unchanged.
- **faults.08** --- (Scanning directory) The directory configured to be scanned for new schedule files does not exist.
- **faults.09** --- (Archiving schedule) The directory configured to receive archived schedule file does not exist or moving a schedule file to this directory failed for some reason (medium full, missing file access privileges). In this case the old schedule file gets deleted rather than archived.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.antennaId	TEXT	SAVE SETUP	StringRange
config.scheduleDir	TEXT	SAVE SETUP	StringRange
config.archiveDir	TEXT	SAVE SETUP	StringRange
state.validFrom	TEXT	R/O	StringRange R/O
state.validTo	TEXT	R/O	StringRange R/O
state.generated	TEXT	R/O	StringRange R/O
state.originator	TEXT	R/O	StringRange R/O
state.requestReference	TEXT	R/O	StringRange R/O
state.scheduleName	TEXT	R/O	StringRange R/O
state.satelliteName	TEXT	R/O	StringRange R/O
state.requestedAntenna	TEXT	R/O	StringRange R/O
state.remark	TEXT	R/O	StringRange R/O
state.activeConfig	TEXT	R/O	StringRange R/O
state.activePass	TEXT	R/O	StringRange R/O
state.activeAntenna	TEXT	R/O	StringRange R/O
state.activeSatellite	TEXT	R/O	StringRange R/O
state.passList	TEXT	R/O	StringRange R/O
state.passList.coldef	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O
state.enabled	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
state.phase	TEXT	R/O	StringRange R/O
running	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
withinPass	BOOLEAN	R/O	BooleanRange R/O (true false)
disable	TEXT	NOPRESET	StringRange
interrupt	TEXT	NOPRESET	StringRange
enable	TEXT	NOPRESET	StringRange
delete	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
mem.validFrom	INTEGER	SAVE	IntegerRange (0 .. 0)
mem.validTo	INTEGER	SAVE	IntegerRange (0 .. 0)
mem.generated	INTEGER	SAVE	IntegerRange (0 .. 0)
mem.originator	TEXT	SAVE	StringRange
mem.requestReference	TEXT	SAVE	StringRange
mem.scheduleName	TEXT	SAVE	StringRange
mem.satelliteName	TEXT	SAVE	StringRange
mem.requestedAntenna	TEXT	SAVE	StringRange
mem.remark	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Events missed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Events dropped)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Schedule load)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pass outside)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Swapped pass times)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Overlapping passes)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Overlaps current time)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Scanning directory)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Archiving schedule)

9.516 Peak-Beacon-Receiver

This page describes the device driver and the device window for the Peak Beacon Receiver included in UPC7000 series (Automatic Up-Link Power Control).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit as well as for initial setup.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters

recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- loEnable --- Defines if an frequency offset should be used for the beacon rx frequency.
- loFreq --- Defines the lo frequency offset in MHz.
- loInversion --- Defines if rx frequency should be inverted from LO.
- lnbVolt --- Defines if LNB supply voltage should be switched on/off. ctc
- output10Mhz --- Defines if a 10MHz reference signal should be fed to the LNB.
- antiSideBand --- Defines if the unit should use its single sweep digital anti-sideband detector to prevent the unit locking to beacon sidebands.
- pad10db --- Defines if the unit use its 10dB pad to allow measurements to be better made if the unit has a high aggregate input level.
- vOutOffs --- Defines the beacon log offset, used to trim the output voltage.
- vOutScale --- Defines the scale for the beacon log output voltage:
  - 0.5db/V,
  - 1db/V,
  - 2db/V,
  - 5db/V or
  - 10db/V.
- vOutLowLimit --- Sets the Low voltage limit.
- vOutLHighLimit --- Sets the high voltage limit.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.loEnable	CHOICE	SETUP	EnumRange (ON OFF)
config.loFreq	FLOAT	SETUP	DoubleRange (0 .. 0)
config.loInversion	CHOICE	SETUP	EnumRange (INVERTED NORMAL)
config.lnbVolt	CHOICE	SETUP	EnumRange (ON OFF)
config.output10Mhz	CHOICE	SETUP	EnumRange (ON OFF)
config.pad10db	CHOICE	SETUP	EnumRange (ON OFF)
config.antiSideBand	CHOICE	SETUP	EnumRange (ON OFF)
config.vOutOffs	INTEGER	SETUP	IntegerRange (0 .. 100)
config.vOutScale	CHOICE	SETUP	EnumRange (0.5db/V 1db/V 2db/V 5db/V 10db/V)
config.vOutLowLimit	CHOICE	SETUP	EnumRange (0V 5V 10V)
config.vOutHighLimit	CHOICE	SETUP	EnumRange (0V 5V 10V)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
power	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
gain	FLOAT		DoubleRange (0.0 .. 20.0)
refLevel	INTEGER		IntegerRange (-100 .. 0)
frequency	FLOAT		DoubleRange (950.000 .. 2150.000)
cn.frequency	FLOAT		DoubleRange (950.000 .. 2150.000)
bandwidth	INTEGER		IntegerRange (80 .. 25000)
resolutionBw	CHOICE		EnumRange (1kHz 6kHz)
sweepRate	CHOICE		EnumRange (2.5kHz/s 5kHz/s 10kHz/s 20kHz/s 40kHz/s 80kHz/s 120kHz/s 240kHz/s)
sweepWidth	CHOICE		EnumRange (20kHz 50kHz 100kHz 200kHz 500kHz)
internal.unit	CHOICE		EnumRange (kHz MHz)
internal.bandwidth	INTEGER		IntegerRange (0 .. 0)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Level threshold)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Frequency track)

9.517 Peak-Blockdownconverter

This page describes the device driver and the device window for the for the Peak Blockdownconverter Series via SNMP.

It supports 1+1 and 2+1 configurations .

The driver was tested with **IBDH725** without redundancy **SW: 3.50**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters

recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- lowGain --- Allows to adjust the lower gain limit of the respective unit.
- highGain --- Allows to adjust the upper gain limit of the respective unit.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- priority --- Allows to adjust the priority unit in a redundant setup.
- hasRedundancy --- Define if unit is part of a redundant setup.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.redId	TEXT	R/O	StringRange R/O
info.redType	CHOICE	R/O	EnumRange R/O (1+1 2+1 N+1)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.priority	CHOICE	SETUP	EnumRange (1 2 3 4 5 6 7 8 N/A)
config.hasRedundancy	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange (A B)
state	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
protection	CHOICE		EnumRange (AUTO MANUAL)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Summary)
internal.deltaGain	FLOAT		DoubleRange (0.0 .. 0.0)
internal.gain	FLOAT		DoubleRange (0.0 .. 0.0)

9.518 Peak-Blockupconverter

This page describes the device driver and the device window for the for the Peak Blockupconverter Series via SNMP.

It supports 1+1 and 2+1 configurations.

The driver was tested with **IBUH** without redundancy and UPC **SW: 4.55**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- lowGain --- Allows to adjust the lower gain limit of the respective unit.
- highGain --- Allows to adjust the upper gain limit of the respective unit.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- priority --- Allows to adjust the priority unit in a redundant setup.
- hasRedundancy --- Define if unit is part of a redundant setup.
- hasUpc --- Define if unit is equipped with a UPC.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.redId	TEXT	R/O	StringRange R/O
info.redType	CHOICE	R/O	EnumRange R/O (1+1 2+1 N+1)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.priority	CHOICE	SETUP	EnumRange (1 2 3 4 5 6 7 8 N/A)
config.hasRedundancy	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.hasUpc	CHOICE	SAVE SETUP	EnumRange (NO YES)
tx.on	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)

name	*type*	*flags*	*range*
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange (A B)
state	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
protection	CHOICE		EnumRange (AUTO MANUAL)
priority	CHOICE		EnumRange (1 2 3 4 5 6 7 8 N/A)
reset	TEXT	NOPRESET	StringRange
upc.mode	CHOICE	R/O	EnumRange R/O (OFF AUTO MANUAL)
upc.channel	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
upc.attn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
upc.Range	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (UPC)
internal.deltaGain	FLOAT		DoubleRange (0.0 .. 0.0)
internal.gain	FLOAT		DoubleRange (0.0 .. 0.0)

9.519 Peak-DBU200

This page describes the device driver and the device window for the Peak 1:1 System Controller via SNMP using the Peak DBU200.mib.

The driver was tested with the following SW Releases:

- SW: 1.40

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device parameters
- --- Displays meter readings, no values available for DBU200.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.reference.1	CHOICE		EnumRange (OFF ON UNKNOWN)
info.reference.2	CHOICE		EnumRange (OFF ON UNKNOWN)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (A B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Summary)
internal.pol1	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
internal.pol2	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)

9.520 Peak-DBU200H

This page describes the device driver and the device window for the Peak 1:1 System Controller via SNMP using the Peak DXXH.mib.

The driver was tested with the following SW Releases:

- SW: 1.17

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device parameters
- --- Displays meter readings, no values available for DBU200H.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- copySettings --- Defines if settings are copied during redundancy switching

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.unit.1	TEXT	R/O	StringRange R/O
info.unit.2	TEXT	R/O	StringRange R/O
info.reference.1	CHOICE		EnumRange (LOCKED UNLOCKED N/A)
info.reference.2	CHOICE		EnumRange (LOCKED UNLOCKED N/A)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.copySettings	CHOICE	SAVE SETUP	EnumRange (YES NO)
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (A B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Protection)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Summary)
internal.pol1	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)
internal.pol2	CHOICE	R/O	EnumRange R/O (MAIN STANDBY)

9.521 Peak-Downconverter-7000

This page describes the device driver and the device window for Peak Downconverters 7000er Series via SNMP.

The driver was tested with the following models:

- P7025A SW: 5.07 in 2:1 configuration
- P7001 SW: 5.03 in 1:1 configuration

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- Displays meter readings
- --- The device redundancy parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write

community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- spectrumSettings --- Defines if settings are copied during redundancy switching
- priority --- Defines the priority unit for 2+1 and higher N+1 configurations

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.redId	TEXT	R/O	StringRange R/O
info.redType	CHOICE	R/O	EnumRange R/O (1+1 2+1 N+1)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.priority	CHOICE	SETUP	EnumRange (1 2 3 4 5 6 7 8 N/A)

name	*type*	*flags*	*range*
rx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
rx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
location	CHOICE		EnumRange (A B)
state	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
protection	CHOICE		EnumRange (AUTO MANUAL)
reset	TEXT	NOPRESET	StringRange
rx.invert	CHOICE		EnumRange (NORMAL INVERTED)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (UPC)
internal.deltaGain	FLOAT		DoubleRange (0.0 .. 0.0)
internal.attn	FLOAT		DoubleRange (0.0 .. 0.0)

9.522 Peak-IBDH

This page describes the device driver and the device window for the Peak Redundancy Controller via SNMP. It supports 1:1 and 2:1 configurations.

The driver was tested in 1:1 configuration with the following SW Releases:

- SW: 3.29

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device switch parameters
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- priority --- Defines the priority unit for 2+1 and higher N+1 configurations

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.redId	TEXT	R/O	StringRange R/O
info.redType	CHOICE	R/O	EnumRange R/O (1+1 2+1 N+1)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.priority	CHOICE	SETUP	EnumRange (1 2 3 4 5 6 7 8 N/A)
location	CHOICE		EnumRange (A B)
state	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
protection	CHOICE		EnumRange (AUTO MANUAL)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.523 Peak-P700

This page describes the device driver and the device window for the Peak P700 series Up- and Downconverters, but the driver supports currently only the P702 upconverter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational upconverter parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel or 0 in RS232 mode.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *9600,N,8,1* .

Communication protocol / Interface type

This device driver requires the Peak protocol. In RS232 mode the device address is fixed to 0 (zero, decimal value). In RS485 mode you can define an address.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software. At the front panel, go to 2. Change -> 3. Rem Control -> and select RS232 or RS485 regarding your cabling. Then select the baud rate (e.g. 9600)

To set the device in remote mode, select 2. Change -> 3. Rem Control -> 1. Disable one time. Selecting 1. Disable again, sets the unit back to front panel control.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.status	CHOICE		EnumRange (unit not configured unit configured)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.loFreq	FLOAT	SETUP	DoubleRange (1750.000 .. 13300.000)
config.enableLoFreq	CHOICE	SETUP	EnumRange (ON OFF)
config.buc.loss	INTEGER	SETUP	IntegerRange (0 .. 100)
config.buc.enableLoss	CHOICE	SETUP	EnumRange (ON OFF)
config.buc.10MHz	CHOICE	SETUP	EnumRange (ON OFF)
config.buc.power	CHOICE	SETUP	EnumRange (ON OFF)
tx.on	CHOICE		EnumRange (OFF ON)
tx.input.level	CHOICE		EnumRange (-29dBm -26dBm -23dBm -20dBm -19dBm -16dBm -13dBm -10dBm)
tx.output.level	CHOICE		EnumRange (0dBm -3dBm -6dBm -9dBm -10dBm -13dBm -16dBm -19dBm -20dBm)
tx.frequency	FLOAT		DoubleRange (12750.000 .. 14500.000)
internal.lbandfreq	FLOAT		DoubleRange (0.000 .. 0.000)
internal.f1	TEXT	R/O	StringRange R/O
internal.f2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (-5 volts)
faults.04	ALARM	R/O	AlarmFlagRange R/O (+5 volts)
faults.05	ALARM	R/O	AlarmFlagRange R/O (+15 volts)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Tx 10MHz ref)
faults.07	ALARM	R/O	AlarmFlagRange R/O (L-Band synth)
faults.08	ALARM	R/O	AlarmFlagRange R/O (UHF synth)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Tx temp range)

9.524 Peak-UPC7002

This page describes the device driver and the device window for the Peak Uplink Power Control Unit 7002.

The driver was tested with the following models:

- 2-Channel model UPC 7002 SW 1.60

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
- channels --- Defines the number of channels controlled by the UPC. Up to 10 channels can be used.

- clearSkyCalibrated --- Choosing CALIBRATED will enter the detected power level setting into the saved clear sky value.
- extChanCalibrated --- External channel calibrated allows information to be passed to connected units to correctly display UPC functionality.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.channels	INTEGER	SAVE SETUP	IntegerRange (1 .. 10)
config.clearSkyCalibrated	CHOICE	SETUP	EnumRange (CALIBRATED NOT CALIBRATED)
config.extChanCalibrated	CHOICE	SETUP	EnumRange (CALIBRATED NOT CALIBRATED)
config.hasRedundancy	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.beaconUnit	CHOICE	SAVE SETUP	EnumRange (dBm V)

name	*type*	*flags*	*range*
config.hasFadeLimit	CHOICE	SAVE SETUP	EnumRange (YES NO)
ctrlMode	CHOICE		EnumRange (LOCAL REMOTE)
protection	CHOICE		EnumRange (AUTO MANUAL)
position	CHOICE		EnumRange (A B UNKNOWN)
onlineUnit	CHOICE	R/O	EnumRange R/O (ONLINE OFFLINE)
onlineRed	CHOICE	R/O	EnumRange R/O (ONLINE OFFLINE)
unitIdent	TEXT	R/O	StringRange R/O
position.coax	CHOICE	R/O	EnumRange R/O (A B UNKNOWN)
upcEnable	CHOICE		EnumRange (OFF ON)
inputPowerClearSky	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inputPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
range	FLOAT		DoubleRange (2.0 .. 30.0)
ratio	FLOAT		DoubleRange (0.1 .. 10.0)
stepSize	CHOICE		EnumRange (0.1dB 0.2dB 0.5dB 1dB 2dB)
slewRate	CHOICE		EnumRange (OFF 1dB/10s 1dB/20s 1dB/50s 1dB/100s)
sampleTime	FLOAT		DoubleRange (0.2 .. 10.0)
attn	FLOAT		DoubleRange (0.0 .. 30.0)
hysteresis	FLOAT		DoubleRange (0.0 .. 3.0)
clearSkyOffs	FLOAT		DoubleRange (0.0 .. 4.0)
fadeLimit	FLOAT		DoubleRange DIS (0.0 .. 0.0)
fadeHysteresis	FLOAT		DoubleRange DIS (0.0 .. 0.0)
channel.01	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.02	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.03	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.04	CHOICE		EnumRange DIS (OFF ON DISABLED)

name	*type*	*flags*	*range*
channel.05	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.06	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.07	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.08	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.09	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.10	CHOICE		EnumRange DIS (OFF ON DISABLED)
channel.01.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.02.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.03.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.04.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.05.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.06.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.07.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.08.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.09.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
channel.10.type	CHOICE	R/O	EnumRange R/O DIS (NOT-USED INTERNAL EXTERNAL)
attnOffs.01	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.02	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.03	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.04	FLOAT		DoubleRange DIS (0.0 .. 20.0)

name	*type*	*flags*	*range*
attnOffs.05	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.06	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.07	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.08	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.09	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attnOffs.10	FLOAT		DoubleRange DIS (0.0 .. 20.0)
attn.01	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.02	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.03	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.04	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.05	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.06	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.07	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.08	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.09	FLOAT		DoubleRange DIS (0.0 .. 10.0)
attn.10	FLOAT		DoubleRange DIS (0.0 .. 10.0)
internal.name	TEXT	R/O	StringRange R/O
internal.psu1	TEXT	R/O	StringRange R/O
internal.psu2	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (UPC Off)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ch 1 UPC off)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Ch 2 UPC off)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ch 3 UPC off)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ch 4 UPC off)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Ch 5 UPC off)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Ch 6 UPC off)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Ch 7 UPC off)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Ch 8 UPC off)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ch 9 UPC off)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Ch 10 UPC off)
faults.12	ALARM	R/O	AlarmFlagRange R/O (UPC Pwr incr limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Summary alarm)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Remote access)

9.525 Peak-Upconverter-7000

This page describes the device driver and the device window for Peak Upconverters 7000er Series via SNMP.

The driver was tested with the following models:

- P7013B SW: 5.07 in 1:1 configuration
- P7002W SW: 5.03 in 1:1 configuration

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device control parameters
- --- Displays meter readings
- --- The device redundancy parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- spectrumSettings --- Defines if settings are copied during redundancy switching
- priority --- Defines the priority unit for 2+1 and higher N+1 configurations

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.version	TEXT	R/O	StringRange R/O
info.redId	TEXT	R/O	StringRange R/O
info.redType	CHOICE	R/O	EnumRange R/O (1+1 2+1 N+1)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.priority	CHOICE	SETUP	EnumRange (1 2 3 4 5 6 7 8 N/A)
tx.on	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
tx.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
location	CHOICE		EnumRange (A B)
state	CHOICE	R/O	EnumRange R/O (ONLINE STANDBY)
protection	CHOICE		EnumRange (AUTO MANUAL)
priority	CHOICE		EnumRange (1 2 3 4 5 6 7 8 N/A)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (UPC)
internal.deltaGain	FLOAT		DoubleRange (0.0 .. 0.0)
internal.attn	FLOAT		DoubleRange (0.0 .. 0.0)

9.526 Pico-EnviroMon-EL005

This page describes the device driver and the device window for the*Pico EnviroMon EL005* temperature logger. The sat-nms device driver for the EL005 logger is capable to monitor up to the first 8 temperature sensors connected. The driver simply provides the live readings of the temperature sensors, it operates the EL005 as a simple 8-channel temperature sensor.

The driver does not provide any means to configure or setup the EL005 device. This has to be done using the EnviroMon software coming with the logger before it is connected to the sat-nms system.

The internal logging capabilities of the EL005 can be used in parallel while the EL005 is connected to the sat-nms M&C. However, to read out the stored readings from the EL005, it's RS232 interface must be disconnected from the sat-nms M&C and connected to a computer running the Pico EnviroMon software. For a more comfortable logging of the temperature readings we recommend the usage of the File-Recorder logical device in the sat-nms software.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual temperature values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which are used to configure the device driver.

- *parameter --- description*
- address --- Doesn't matter. The communication protocol to the temperature logger uses a fixed address (RS232 point-to-point communication), ignores any address entered here.
- sensors --- Select the number of temperature sensors (1..8) to be monitored. The number set here must be less or equal the number of sensors configured at the logger. It is possible to configure 10 sensors at the logger, but only monitor the first 5 of them with

the sat-nms Software. If the driver is set to monitor sensors which are not configured at the logger, an arbitrary temperature value will be displayed for the non-existent sensors.

- name.0 ... name.7 --- You may enter a descriptive name for the location of each sensor (numbered 0..7) or the meaning of the temperature value. A name/description set here is not an equivalent of the name configured for the sensor with the EnviroMon software. The sat-nms Software is not capable to read or set the EnviroMon sensor descriptions. The names of sensors which are not enabled (beyond the number of sensors selected with the 'sensors' configuration parameter), are displayed with a gray background an cannot be set.
- scale.0 ... scale.7 --- With these parameters you can adjust the decimal places for each individual sensor (numbered 0..7). The EnviroMon logger sends the actual temperature as an integer number in hundreds, tenths or full centigrade, depending on the decimal places configured for the individual sensor in the EnviroMon software. You have to set the scaling for each sensor in order to make the value appear correctly:

decimal places set in EnviroMon	***scaling***
##	1
##.#	0.1
##.##	0.01

The scaling values of sensors which are not enabled (beyond the number of sensors selected with the 'sensors' configuration parameter), are displayed with a gray background and cannot be set.

- Serial --- The EL005 logger operates at 57600 baud, 8 bits, no parity, no handshake.

Configuration hints

- To make the EL005 logger work with the sat-nms software, it first must be setup using the EnviroMon software coming with the device.
- All temperature sensors should be configured to record Centigrade. If a sensor is configured to record Fahrenheit, the sat-nms software will show the correct (Fahrenheit) numeric value followed by a "C" for Centigrade.
- You only can use temperature sensors with a EL005 monitored by the sat-nms software. Mixing temperature and alarm circuit sensors breaks the data structure sent to the M&C, the sat-nms software will display nonsense in this case.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.name.0	TEXT	SAVE SETUP	StringRange
config.scale.0	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.0	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.0	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.1	TEXT	SAVE SETUP	StringRange
config.scale.1	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.1	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.1	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.2	TEXT	SAVE SETUP	StringRange
config.scale.2	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.2	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.2	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.3	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.scale.3	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.3	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.3	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.4	TEXT	SAVE SETUP	StringRange
config.scale.4	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.4	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.4	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.5	TEXT	SAVE SETUP	StringRange
config.scale.5	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.5	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.5	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.6	TEXT	SAVE SETUP	StringRange
config.scale.6	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.6	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.6	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.name.7	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.scale.7	CHOICE	SAVE SETUP	EnumRange (1 0.1 0.01)
config.minTemp.7	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.maxTemp.7	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.sensors	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8)
internal.0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature.0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.5	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.6	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature.7	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (temperature 1 LOW)
faults.02	ALARM	R/O	AlarmFlagRange R/O (temperature 1 HIGH)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (temperature 2 LOW)
faults.04	ALARM	R/O	AlarmFlagRange R/O (temperature 2 HIGH)
faults.05	ALARM	R/O	AlarmFlagRange R/O (temperature 3 LOW)
faults.06	ALARM	R/O	AlarmFlagRange R/O (temperature 3 HIGH)
faults.07	ALARM	R/O	AlarmFlagRange R/O (temperature 4 LOW)
faults.08	ALARM	R/O	AlarmFlagRange R/O (temperature 4 HIGH)
faults.09	ALARM	R/O	AlarmFlagRange R/O (temperature 5 LOW)
faults.10	ALARM	R/O	AlarmFlagRange R/O (temperature 5 HIGH)
faults.11	ALARM	R/O	AlarmFlagRange R/O (temperature 6 LOW)
faults.12	ALARM	R/O	AlarmFlagRange R/O (temperature 6 HIGH)
faults.13	ALARM	R/O	AlarmFlagRange R/O (temperature 7 LOW)
faults.14	ALARM	R/O	AlarmFlagRange R/O (temperature 7 HIGH)
faults.15	ALARM	R/O	AlarmFlagRange R/O (temperature 8 LOW)
faults.16	ALARM	R/O	AlarmFlagRange R/O (temperature 8 HIGH)

9.527 Pointing-Lock

The Pointing-Lock logical device checks the lock state of up to twenty IRD-Chain devices and controls the enable state of one Antenna-Pointing device.

The function of the Pointing-Lock device is to disable the antenna operation if at least one of

the configured IRD-Chains has selected this antenna AND is set to 'locked' state. This prevents the operator from moving the antenna while it is used to receive a program. The chains may be represented by IRD-Chain or RX-Channel-Select-CDB devices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Pointing-Lock device with the antenna controller and the IRD-Chain devices.

parameter	***description***
pointingDevice	The name of the Antenna-Pointing device which controls the antenna of interest. The Pointing-Lock disabled this device by setting it's mode to OUT-OF-SERVICE. For this reason you may not use a Antenna-Tracking device to control the antenna, the device would suspend tracking while it is disabled. pointingDevice may be left empty if no Antenna-Pointing device shall be controlled (the LOCK state ist still signalled via the 'locked' state variable of the Pointing-Lock device.
antennaName	The name of the antenna as it appears in the configuration file / drop-down box of the IRD-Chain devices. Note that all IRD-Chain devices must use the same name for a particular antenna to let the Pointing-Lock device work properly.
irdChainDevice.??	These parameters define the names of the IRD-Chain or RX-Channel-SElect-CDB devices the Pointing-Lock shall monitor. The parameters may be assigned in arbitrary order, empty / unused entries are skipped by the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.pointingDevice	TEXT	SAVE SETUP	StringRange
config.antennaName	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.1	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.2	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.3	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.4	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.5	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.6	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.7	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.8	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.9	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.10	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.11	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.12	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.13	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.14	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.15	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.16	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.17	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.18	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.19	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.20	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.21	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.22	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.23	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.24	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.25	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.irdChainDevice.26	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.27	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.28	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.29	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.30	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.31	TEXT	SAVE SETUP	StringRange
config.irdChainDevice.32	TEXT	SAVE SETUP	StringRange
locked	BOOLEAN	R/O	BooleanRange R/O (true false)

9.528 Pointing-Robot

This page describes the device driver and the device window for the Pointing-Robot logical device. This device lets you make a motorized antenna follow a satellite based on it's (known) orbit position. For this the Pointing-Robot implements orbit position calculations based on common ephemeris data models. The basic functions of a Pointing-Robot device are:

- Point the antenna to the nominal position of a satellite.
- Move the antenna along a satellite position calculated from ephemeris data.
- Move the antenna along a path given by a table.

The Pointing-Robot is part of the sat-nms ACU Indoor Unit. It supplements the sat-nms ACU-ODM with the orbit calculation routines described below. The Pointing-Robot principally works together with all drivers for antenna controllers which are part of the sat-nms software. However some functions like automatically switching off the ACU's tracking facility are uniquely designed for the sat-nms ACU-ODM.

Move the antenna along a satellite position calculated from ephemeris data

You can make the pointing robot follow a satellite position computed from usual ephemeris data sets. The robot recognizes the following data formats:

- **Keplerian elements** --- The robot recognizes and parses keplerian elements in the NASA 2-line format. The two lines of element definition may be prepended by an additional line with the satellite's name. A common source for Keplerian elements in this format is www.celestrak.com . The pointing robot uses the NORAD SDP4 model to predict a satellite's position from this type of data.
- **Intelsat elements** --- Intelsat publish ephemeris data for their communication satellites at their web site. The data format and the mathematical model for this data is Intelsat

proprietary. The pointing robot recognizes this data type as well. You should copy the complete text of the ephemeris data set into the entry field to make the software recognize the data correctly.

Pointing the antenna to the nominal position of a satellite

If a single floating point number is entered to the ephemeris data field, the pointing robot interprets this as a nominal orbit position of the satellite (°E).

Move the antenna along a path given by a table

The pointing robot is capable to move the antenna along a path defined in a text file table. You either may copy the file to the IDU computer by means of FTP and enter the name of file to the ephemeris field or you may enter or cot/paste the table directly to this field. If you do the latter, the size of the table is limited to 32000 characters.

The Pointing-Robot provides a file selection dropdown widget to select files located in the /home/satnms/robot directory on the M&C&VLC server. To use this feature, select the file from the dropdown list and then click "SELECT". This copies the file name into the ephemeris data field and thus selects the file.

Files in the /home/satnms/robot directory which are no longer used may be deleted by selecting the file do delete from the drop down list and clicking DELETE. A file actually in use cannot be deleted, the request is silently ignored.

Polarization calculation

When calculating the antenna pointing from a orbital model or when interpolating the pointing from the table data, the Pointing-Robot also calculates a polarization angle. As each satellite uses its individual polarization offset, the calculated angle is normally for information only.

With table tracking, the pointing robot may also command the calculated polarization angle at the antenna. Setting "apply pol" to YES enables this feature. "apply pol" is not available with orbital elements.

Tracking table format

The format of the table must be as follows: It contains the data as a three or four column table. Empty lines are ignored, comments starting with a '#' as well. The numbers in the table are parsed as floating point numbers which only may consist of decimal digits, one decimal point and an optional leading '-' if negative. The columns must be separated by an arbitrary number of space or tabulator characters. They have the following meanings:

- 1 --- time stampThe time stamp must be a Julian date with the time of day coded as a fraction of a day. Example: The Julian date for the common base of most computer clocks (1970-01-0100:00:00) is 2440587.5
- 2 --- azimuth angleThe azimuth angle in degrees
- 3 --- elevation angleThe elevation angle in degrees

File import

The Pointing Robot GUI offers an import function which permits to copy pointing tables from the client computer to the server and thereby convert the file format to that one internally used by the sat-nms software.

Parsing order

If the pointing robot gets entered a new ephemeris data set, it tries to interpret the data in the following way:

1. It checks if the first line of the data specifies an existing file (residing on the IDU computer, therefore the file path should use Unix-style ('/') directory separators if necessary. If there is the specified file, it gets read an interpreted in a way, its file extension suggests: '*.tle' or '*.TLE': NASA 2-line ephemeris data. '*.i11' or '*.I11': Intelsat ephemeris data. Files with other extensions are interpreted as table data as described above.
2. It checks if the data is a valid NASA 2-line data set. A satellite name before the 2-line data is accepted and parsed separately if present.
3. It checks if the data contains valid Intelsat ephemeris data.
4. It checks if the first number in the first line of the data is a valid Julian date in the range 2005 .. 2050. In this case the robot assumes, that the ephemeris data contain a pointing table.
5. Finally, if nothing of the above mentioned formats did match, the robot tries to interpret the first line of the data as the nominal orbit position of the satellite.

If all of the parsing attempts fail, the pointing robot reports the model type to be UNKNOWN. Mostly this happens due to artifacts of previously entered elements which had not been deleted before the new ephemeris data was entered / pasted.

Remote control

The Pointing-Robot may be updated easily with updated pointing tables or ephemeris data by copying such files to the /home/public directory of the M&C system using the SMB file sharing protocol and then commanding the name of this file to the 'elements.input' variable of the Pointing-Robot using one of the M&C system's remote control ports.

ACU fault monitoring

The device can monitor the ACU's summary fault and show it in the device screen. The parameter 'antenna fault detection' may be set to one of the following settings:

- **IGNORE-ACU-FAULT** : The pointing robot ignores ACU faults, does not show them.
- **SHOW-ACU-FAULT** : The pointing robot shows ACU faults, continues to command the antenna controller even if there is an ACU fault.
- **STOP-ON-ACU-FAULT** : The pointing robot stops when it detects a fault of the antenna controller. It must be restarted manually after the ACU fault has been cleared.

Fault flags

In table tracking mode the Pointing-Robot reports two fault conditions regarding the end of table data:

File expired is reported if the end of the table data is reached. Some hours before (configurable in the device setup) the **File expires soon** fault is reported. The latter fault condition appears as a WARNING by default.

Both fault conditions are reported if the Pointing-Robot is in table tracking mode, even if the tracking is stopped. To clear faults concerning an outdated but unused table, simply clear the "ephemeris data" field.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
antennaName	The name of the antenna ODM device to control. With the sat-nms IDU, this parameter is preconfigured to 'ODM' and should not be changed.
longitude	The antenna's longitude (°E).
latitude	The antenna's latitude (°N).
altitude	The antenna's altitude over sea (m)
expireWarning	Defines with table tracking how many hours before the end of data a "Files expires soon" warning / fault shall be rise.

Remarks

The accuracy of the calculations made by Pointing-Robot depend on the following:

- The geodetic location of the antenna must be known very precisely.
- The antenna's azimuth axis alignment must be precisely perpendicular.
- The clock of the M&C computer must be set to UTC.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.antennaName	TEXT	SAVE SETUP	StringRange
config.longitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.altitude	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
tracking	CHOICE	SAVE	EnumRange (OFF ON)
interval	INTEGER	SAVE	IntegerRange (0 .. 3600)
calculatePol	CHOICE	SAVE	EnumRange (NO YES)
antFaultMode	CHOICE	SAVE	EnumRange (IGNORE-ACU-FAULT SHOW-ACU-FAULT STOP-ON-ACU-FAULT)
expireWarning	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
elements.input	TEXT		StringRange
eval.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
eval.orbit	TEXT	R/O	StringRange R/O
azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
polarization	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.activeFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.latchedFault	BOOLEAN	R/O	BooleanRange R/O (true false)
state.short	TEXT	R/O	StringRange R/O
state.long	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
offset.az	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.el	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
offset.pl	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
hysteresis.az	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.el	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
hysteresis.pl	FLOAT	SAVE	DoubleRange (0.000 .. 10.000)
elements.type	TEXT	R/O	StringRange R/O
elements.age	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
elements.epoch	TEXT	R/O	StringRange R/O
elements.validFrom	TEXT	R/O	StringRange R/O
elements.validTo	TEXT	R/O	StringRange R/O
evalByElements	TEXT	NOPRESET	StringRange
moveEvaluated	TEXT	NOPRESET	StringRange
resetLatched	TEXT	NOPRESET	StringRange
files	CHOICE		EnumRange (NONE)
selectFile	TEXT	NOPRESET	StringRange
deleteFile	TEXT	NOPRESET	StringRange
forceUpdateFileList	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (File expired)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (File expires soon)

9.529 Polarization-Select

The *Polarization-Select* logical device lets you derive a polarization plane information from the antenna's polarization angle value. The other way round it lets you command a polarization plane by turning the feed by 90° to the on or other direction.

The device uses a simple algorithm to determine the polarization plane from the angle: Within the possible range of polarization angle settings there is at least one threshold which separates horizontal from vertical settings. Switching the polarization is made by shifting the angle by 90° across this threshold.

By definition the polarization plane controlled by the device in this way is the receive polarization. Additionally the device provides the orthogonal setting as a read-only parameter labelled 'transmit-polarization'.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Monitors / controls the polarization plane.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	*description*
polarizationId	The parameter ID of the polarization angle of the antenna. This must be the parameter that actually controls the angle, not the state report parameter.
separationMode	Selects which range of the polarization angle shall be considered as horizontal, which a vertical. 'HORIZONTAL-BELOW' treats values below the threshold as horizontal, values above as vertical. 'HORIZONTAL-ABOVE' sets the opposite sense.

parameter	*description*
separationValue	Sets the threshold value which separates horizontal from vertical settings. If the possible range of polarization angles covers more than 180°, then there are more than one of these separation points. In such a case, a threshold value near to the center of the possible range should be used.

Remarks

The Polarization-Select device cannot handle 360° rollovers of the angle. This is, because then the angle - polarization plane correlation would be no more biunique, making it impossible to reconstruct the original settings after a plane switch.

For this reason you should calibrate / offset the polarization plane setting at the antenna in a way, that not 360° rollovers happen in the valid angle range.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.angle	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.initialized	BOOLEAN	R/O	BooleanRange R/O (true false)
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.separationMode	CHOICE	SAVE SETUP	EnumRange (HORIZONTAL-BELOW HORIZONTAL-ABOVE)
config.separationValue	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
rxPol	CHOICE		EnumRange (UNKNOWN)
txPol	CHOICE	R/O	EnumRange R/O (UNKNOWN X Y)

9.530 PowerCalc-Transponder

The logical device PowerCalc-Transponder, calculates the number of remaining transponders for a given HPA power.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
config.pwrMeasId	The message ID for the output power reading.
config.faultId	The message ID to detect and communication fault of the measurement device
config.offsetId	The message ID for the offset / EIRP calibration factor. Usually taken from the related EIRP device e.g. EIRP.offset
config.maxPower	Defines the maximum linear output power in Watt
config.eirpTp	Required power per Transponder in dBW
config.weatherBackupTp	Number of transponders for weather backup

The device calculates the actual number of transponders with the current measured output power and the number of remaining transponders which could be added to the HPA without exceeding the maximal available output power.

If the actual output power exceeds this maximum power the device will raise a fault.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.faultId	TEXT	SAVE SETUP	StringRange
config.offsetId	TEXT	SAVE SETUP	StringRange
config.maxPower	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.eirpTp	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.weatherBackupTp	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
measuredEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
maxEirp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
actualTp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
remainingTp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.meas	INTEGER	NOPRESET	IntegerRange (0 .. 0)
internal.flt	BOOLEAN	NOPRESET	BooleanRange (true false)
internal.offset	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power exceeded)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Meas. Device Comm.)

9.531 PowerCalc

The logical device PowerCalc, calculates the overall output power of two HPAs and take into account the position of a 2+1 redundancy switch and if a power combiner (VPC) is active or not.

Device window pages

The following table shows which device window pages are available with this individual device

type.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
hpa1Id	Parameter ID of first HPA output power in W
hpa2Id	Parameter ID of second HPA output power in W
hpaRId	Parameter ID of redundant HPA output power in W
hpa1Id	Parameter ID of redundancy switch position
hpa1Id	Parameter ID of variable power combiner position

The result is provided via pOut (in Watt) and p (in dBW) parameter.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.hpa1Id	TEXT	SAVE SETUP	StringRange
config.hpa2Id	TEXT	SAVE SETUP	StringRange
config.hpaRId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.red	TEXT	SAVE SETUP	StringRange
config.vpc	TEXT	SAVE SETUP	StringRange
pHpa1	INTEGER		IntegerRange (0 .. 0)
pHpa2	INTEGER		IntegerRange (0 .. 0)
pHpaR	INTEGER		IntegerRange (0 .. 0)
p1	INTEGER		IntegerRange (0 .. 0)
p2	INTEGER		IntegerRange (0 .. 0)
pOut	INTEGER		IntegerRange (0 .. 0)
p	FLOAT		DoubleRange (0.00 .. 0.00)
red	CHOICE		EnumRange (NONE 1 2)
vpc	CHOICE		EnumRange (A B A+B LOAD)

9.532 Profline-Plus1-AudioReceiver

This page describes the *Profline-Plus1-AudioReceiver* device driver and its device window. The *Profline-Plus1-AudioReceiver* device driver monitors and controls the FM and DAB receiver.

The driver was tested with the following models:

- Model: extFM,extDAB,Streamer
- SW: 2.19

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The FM monitoring and control parameters.
- --- The DAB monitoring and control parameters.
- --- The audio decoder output monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
- dabLevelMonitoring --- Defines the DAB rf level alarm monitoring: Off, Relay or Contact1-4.
- dabLevelMode --- Defines the comparator:
    - LT Lower then,
    - LE - Lower or equal then,
    - EQ - Equal,
    - NE - Not equal,
    - GT - Greater then,
    - GE - Greater or equal then.
- dabLevelThreshold --- Defines the threshold level in dBuV (0..120)
- dabLevelTimeout --- Defines the timeout delay before an alarm is raised.
- fmLevelMonitoring --- Defines the FM rf level alarm monitoring: Off, Relay or Contact1-4.
- fmLevelMode --- Defines the comparator:
    - LT Lower then,

  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,
  - GT - Greater then,
  - GE - Greater or equal then.

- fmLevelThreshold --- Defines the threshold level in dBuV (0..120).
- fmLevelTimeout --- Defines the timeout delay before an alarm is raised.
- fmPilotMonitoring --- Defines the FM pilot deviation level alarm monitoring: Off, Relay or Contact1-4.
- fmPilotMode --- Defines the comparator:
  - LT Lower then,
  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,
  - GT - Greater then,
  - GE - Greater or equal then.
- fmPilotThreshold --- Defines the threshold level in kHz (1..15).
- fmPilotTimeout --- Defines the timeout delay before an alarm is raised.
- rdsBerMonitoring --- Defines FM RDS bit error rate alarm monitoring: Off, Relay or Contact1-4.
- rdsBerMode --- Defines the comparator:
  - LT Lower then,
  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,
  - GT - Greater then,
  - GE - Greater or equal then.
- rdsBerThreshold --- Defines the threshold level in % (0..100).
- rdsBerTimeout --- Defines the timeout delay before an alarm is raised.
- rdsDevMonitoring --- Defines the RDS deviation alarm monitoring: Off, Relay or Contact1-4.
- rdsDevMode --- Defines the comparator:
  - LT Lower then,
  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,

    - GT - Greater then,
    - GE - Greater or equal then.

- rdsDevThreshold --- Defines the threshold level in kHz (0..100).
- rdsDevTimeout --- Defines the timeout delay before an alarm is raised.
- tpMonitoring --- Defines the TP (traffic program) value alarm monitoring: Off, Relay or Contact1-4.
- tpMode --- Defines the comparator:

    - EQ - Equal,
    - NE - Not equal.

- tpValue --- Defines the alarm value ON or OFF.
- tpTimeout --- Defines the timeout delay before an alarm is raised.
- taMonitoring --- Defines the TA (traffic announcement) value alarm monitoring: Off, Relay or Contact1-4.
- taMode --- Defines the comparator:

    - EQ - Equal,
    - NE - Not equal.

- taValue --- Defines the alarm value ON or OFF.
- taTimeout --- Defines the timeout delay before an alarm is raised.
- tptaMonitoring --- Defines the TP and TA value alarm monitoring: Off, Relay or Contact1-4.
- tptaMode --- Defines the comparator:

    - EQ - Equal,
    - NE - Not equal.

- tptaValue --- Defines the alarm value ON or OFF.
- tptaTimeout --- Defines the timeout delay before an alarm is raised.
- diMonitoring --- Defines the DI value alarm monitoring: Off, Relay or Contact1-4.
- diMode --- Defines the comparator:

    - EQ - Equal,
    - NE - Not equal.

- diValue --- Defines the alarm value MONO or STEREO.
- diTimeout --- Defines the timeout delay before an alarm is raised.

- audioLeftMonitoring --- Defines the left audio channel level alarm monitoring: Off, Relay or Contact1-4.
- audioLeftMonMode --- Defines the comparator:
  - LT Lower then,
  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,
  - GT - Greater then,
  - GE - Greater or equal then.
- audioLeftThreshold --- Defines the threshold level in dBuV (-72..0).
- audioLeftTimeout --- Defines the timeout delay before an alarm is raised.
- audioRightMonitoring.tc Defines the right audio channel level alarm monitoring: Off, Relay or Contact1-4.
- audioRightMonMode --- Defines the comparator:
  - LT Lower then,
  - LE - Lower or equal then,
  - EQ - Equal,
  - NE - Not equal,
  - GT - Greater then,
  - GE - Greater or equal then.
- audioRightThreshold --- Defines the threshold level in dBuV (-72..0).
- audioRightTimeout --- Defines the timeout delay before an alarm is raised.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.kernel	TEXT	R/O	StringRange R/O
info.display	TEXT	R/O	StringRange R/O
info.board	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.dabLevelMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.dabLevelMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.dabLevelThreshold	INTEGER	SETUP	IntegerRange (0 .. 120)
config.dabLevelTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.fmLevelMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.fmLevelMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.fmLevelThreshold	INTEGER	SETUP	IntegerRange (0 .. 120)
config.fmLevelTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.fmPilotMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.fmPilotMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.fmPilotThreshold	FLOAT	SETUP	DoubleRange (1 .. 15)
config.fmPilotTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)

name	*type*	*flags*	*range*
config.rdsBerMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.rdsBerMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.rdsBerThreshold	INTEGER	SETUP	IntegerRange (0 .. 100)
config.rdsBerTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.rdsDevMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.rdsDevMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.rdsDevThreshold	FLOAT	SETUP	DoubleRange (0.0 .. 10.0)
config.rdsDevTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.tpMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.tpMode	CHOICE	SETUP	EnumRange (EQ NE)
config.tpValue	CHOICE	SETUP	EnumRange (OFF ON)
config.tpTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.taMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.taMode	CHOICE	SETUP	EnumRange (EQ NE)
config.taValue	CHOICE	SETUP	EnumRange (OFF ON)
config.taTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.tptaMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.tptaMode	CHOICE	SETUP	EnumRange (EQ NE)
config.tptaValue	CHOICE	SETUP	EnumRange (OFF ON)
config.tptaTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)

name	*type*	*flags*	*range*
config.diMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.diMode	CHOICE	SETUP	EnumRange (EQ NE)
config.diValue	CHOICE	SETUP	EnumRange (MONO STEREO)
config.diTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.audioLeftMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.audioLeftMonMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.audioLeftThreshold	FLOAT	SETUP	DoubleRange (-72.0 .. 0.0)
config.audioLeftTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.audioRightMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.audioRightMonMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.audioRightThreshold	FLOAT	SETUP	DoubleRange (-72.0 .. 0.0)
config.audioRightTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
config.tempMonitoring	CHOICE	R/O	EnumRange R/O (OFF RELAY CONTACT-1 CONTACT-2 CONTACT-3 CONTACT-4)
config.tempMode	CHOICE	SETUP	EnumRange (LT LE EQ NE GT GE)
config.tempThreshold	INTEGER	SETUP	IntegerRange (0 .. 90)
config.tempTimeout	INTEGER	SETUP	IntegerRange (0 .. 600)
state.fm.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fm.deviation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fm.offset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.fm.pilotDeviation	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.fm.adjacentLow	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fm.adjacentHigh	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
state.fm.multipath	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.rds.deviation	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.rds.ber	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.rds.tp	CHOICE	R/O	EnumRange R/O (ON OFF)
state.rds.ta	CHOICE	R/O	EnumRange R/O (ON OFF)
state.rds.ms	CHOICE	R/O	EnumRange R/O (MUSIC SPEECH)
state.rds.di	CHOICE	R/O	EnumRange R/O (MONO STEREO)
state.rds.pty	TEXT	R/O	StringRange R/O
state.rds.pi	TEXT	R/O	StringRange R/O
state.rds.ps	TEXT	R/O	StringRange R/O
state.rds.rt	TEXT	R/O	StringRange R/O
state.dab.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.snr	TEXT	R/O	StringRange R/O
state.dab.cnr	TEXT	R/O	StringRange R/O
state.dab.fic	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.msc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.cu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.id	TEXT	R/O	StringRange R/O
state.dab.label	TEXT	R/O	StringRange R/O
state.dab.srvLabel	TEXT	R/O	StringRange R/O
state.dab.srvId	TEXT	R/O	StringRange R/O
state.dab.srvCompId	TEXT	R/O	StringRange R/O
state.dab.protection	TEXT	R/O	StringRange R/O
state.dab.srvBitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.dab.samplerate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
state.dab.mode	TEXT	R/O	StringRange R/O
state.dab.ps	CHOICE	R/O	EnumRange R/O (OFF ON)
state.dab.sbr	CHOICE	R/O	EnumRange R/O (OFF ON)
state.dab.drcGain	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.left	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
audio.right	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fm.input	CHOICE		EnumRange (1 2)
fm.frequency	FLOAT		DoubleRange (87.50 .. 108.00)
fm.mute	CHOICE		EnumRange (OFF 1dBuV 2dBuV 3dBuV 4dBuV 5dBuV 6dBuV 7dBuV 8dBuV 9dBuV 10dBuV 11dBuV 12dBuV 13dBuV 14dBuV 15dBuV 16dBuV 17dBuV 18dBuV 19dBuV 20dBuV 21dBuV 22dBuV 23dBuV 24dBuV 25dBuV 26dBuV 27dBuV 28dBuV 29dBuV 30dBuV 31dBuV 32dBuV 33dBuV 34dBuV 35dBuV 36dBuV 37dBuV 38dBuV 39dBuV 40dBuV 41dBuV 42dBuV 43dBuV 44dBuV 45dBuV 46dBuV 47dBuV 48dBuV 49dBuV 50dBuV)
fm.monoStereoThd	INTEGER		IntegerRange (20 .. 50)
fm.deEmphasis	CHOICE		EnumRange (50usec 75usec)
dab.input	CHOICE		EnumRange (1 2)
dab.channel	CHOICE		EnumRange (MANUAL 5A 5B 5C 5D 6A 6B 6C 6D 7A 7B 7C 7D 8A 8B 8C 8D 9A 9B 9C 9D 10A 10B 10C 10D 11A 11B 11C 11D 12A 12B 12C 12D 13A 13B 13C 13D 14A 14B 14C 14D 15A 15B 15C 15D 16A 16B 16C 16D 17A 17B 17C 17D)
dab.frequency	FLOAT		DoubleRange (174.928 .. 260.640)
dab.service	INTEGER		IntegerRange (1 .. 32)

name	*type*	*flags*	*range*
dab.programList	CHOICE		EnumRange (00 - UNKNOWN)
output	CHOICE		EnumRange (DAB FM)
testtone	CHOICE		EnumRange (ON OFF)
audio.gain.ana	INTEGER		IntegerRange (-24 .. 10)
audio.gain.dig	INTEGER		IntegerRange (-24 .. 10)
audio.gain.mpx	FLOAT		DoubleRange (-3.0 .. 3.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (DAB RF Level)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FM RF Level)
faults.03	ALARM	R/O	AlarmFlagRange R/O (FM Pilot)
faults.04	ALARM	R/O	AlarmFlagRange R/O (FM RDS BER)
faults.05	ALARM	R/O	AlarmFlagRange R/O (FM RDS Deviation)
faults.06	ALARM	R/O	AlarmFlagRange R/O (FM RDS TP)
faults.07	ALARM	R/O	AlarmFlagRange R/O (FM RDS TA)
faults.08	ALARM	R/O	AlarmFlagRange R/O (FM RDS TA+TP)
faults.09	ALARM	R/O	AlarmFlagRange R/O (FM RDS DI)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Left Audio)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Right Audio)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature)

9.533 Promax-Sathunter

This page describes the device driver and the device window for the Promax Sathunter device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page controls the receive settings and shows the satellite receive level / satellite identification.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
bandEdge	Low/High band are separated at this frequency (MHz)
loFreqHigh	The high band LO frequency to be used for the frequency calculation.
loFreqLow	The low band LO frequency to be used for the frequency calculation.
Serial	The line parameters for the serial interface expected by the Sathunter device are *115200,N,8,1* .

Communication setup

The Sathunter device is connected to the VLC through an USB interface. This requires that the VLC must be prepared to recognize the Sathunter as a serial USB device.

Before the Sathunter ist connected to the VLC, it should be configured to display the receive level as 'dBm' rather than 'dBuV' which is the Sathunter's shipping configuration. To change this parameter, the Windows based Software coming with the Sathunter device is required.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.ipn	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.version	TEXT	R/O	StringRange R/O
info.fpga	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.bandEdge	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.loFreqLow	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
config.loFreqHigh	FLOAT	SAVE SETUP	DoubleRange (0 .. 0)
internal.xx	HEX	R/O	HexRange R/O (0 .. 0)
internal.yy	HEX	R/O	HexRange R/O (0 .. 0)
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.band	TEXT	R/O	StringRange R/O
satpower	INTEGER	R/O	IntegerRange R/O (0 .. 100)
maxpower	INTEGER	R/O	IntegerRange R/O (0 .. 100)
netid	TEXT	R/O	StringRange R/O
netname	TEXT	R/O	StringRange R/O
orbit	TEXT	R/O	StringRange R/O
services	TEXT	R/O	StringRange R/O
chanpower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
mer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
cber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
vber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
reset	TEXT	NOPRESET	StringRange
detect	TEXT	NOPRESET	StringRange
identify	TEXT	NOPRESET	StringRange
lnbsupply	CHOICE		EnumRange (OFF 13V 13V+22kHz 18V 18V+22kHz)

name	*type*	*flags*	*range*
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
standard	CHOICE		EnumRange (DVB-S DVB-S2)
modulation	CHOICE		EnumRange (QPSK 8PSK)
symbolrate	FLOAT		DoubleRange (0.000 .. 0.000)
fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 1/4 1/3 2/5 3/5 8/9 9/10)

9.534 Protection-Switch-1-To-1-Dual

The Protection-Switch-1-To-1-Dual logical device provides a software based solution for a 1:1 symmetric redundancy switch. The devices knows two basic modes called "SINGLE" and "DUAL". Although designed to switch between two redundant Antennas, the device may be used in other environments as well.

SINGLE mode

In SINGLE mode the device knows one active and one redundant antenna (or e.g. transmit chain) which is in stand-by state. When the active side becomes faulty and the stand-by side shows no fault, the devices switches over to the other side. This works symmetrically, there is no preferred side for the device. Depending on the SW-ONCE/SW-ALWAYS setup parameter, the device locks switching after it switched once until the operator releases this by clicking the RESET button (SW-ONCE). In SW-ALWAYS mode, the device will switch back to the original side if the the switchover condition appears in the opposite sense.

When used with two Antennas, the SINGLE mode is used if all RX- and TX- signals are routed through the active antenna only. The stand-by Antenna is not in use.

DUAL mode

This mode is used with two antennas, which are used at the same time (load sharing). In case of a fault on one side, all signals used with this antenna are routed through the other one. The device does not check a "location" input signalling which antenna actually is in use.

Fault monitoring

The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch over. Actually the device supports five fault flags for each chain which are logically ORed to obtain the chain fault.

Delay pararameters

The Protection-Switch-1-To-1-Dual logical device provides two delay parameters which control the device's behaviour in different situations.

- Switchdelay --- This delay controls the time, a fault condition must be present before the device switches from primary to backup
- Recoverydelay --- If the device switches to backup, the primary chain states OK again and a fault happens at the backup chain, the device switches back to the primary chain after this time. This delay prevents the redundancy switch from oscillating if the switch over process itself causes a short fault at the backup chain.

All delay values may be set to zero resulting in an immediate reaction of the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
operationMode	may be set to SW-ONCE or SW-ALWAYS to control how the device handles consecutive faults.
switchMode	may be set to SINGLE or DUAL to control device's general mode.
fault11Id ... fault15Id	The message IDs for the fault flags to be monitored for the side 1. The device recognizes device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well.
fault21Id ... fault25Id	The message IDs for the fault flags to be monitored for the side 2. The device recognizes device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well.
switchFaultId	The message ID of the fault flag indicating a problem with the chain select switch. Usually this is 'ZZZ.fault', if 'ZZZ' is the device name of the switch.
switchId	The message ID to send the switch over command to. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.

parameter	***description***
location1Val	The command to be sent to 'switchId' if the protection switch wants to select the side 1. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
location2Val	The command to be sent to 'switchId' if the protection switch wants to select the side 2. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
locdetectId	The message ID to monitor in order to detect the actual primary/backup state. This may be (in case of a simple switch) the same as switchId, but if switchId for example triggers a macro to perform a more complex switchover procedure you may monitor another parameter to detect if the primary or the backup chain is selected.
locdetect1Val	The value expected from the parameter addressed by locdetectId for side 1
locdetect2Val	The value expected from the parameter addressed by locdetectId for side 2
suppressFaultIfOff	may set to YES or NO. If set to YES, single fault and double fault are suppressed if protection is set to OFF. If set to NO, these faults are reported even if protection is OFF.

Fault definitions

The Protection-Switch-1-To-1-Dual device defines a couple of individual faults to report the device's actual state. The fault conditions *always* are shown as LEDs at the first page if the device window. Depending on the individual fault configuration, these conditions also may be logged and summarized to a device fault for the redundancy switch. The table below describes the fault flags defined for the Protection-Switch-1-To-1-Dual device.

- Single Fault --- Becomes true if at least one of the configured fault inputs becomes true. This also means that -- if the switch over already has occurred -- the protection switch actually is not able to compensate a possible fault in the remaining chain. Enable this fault if you want to be warned in this case.
- Double Fault --- In each chain at least one device reports a fault. The (transmit-) facility probably is not operable in this case.
- Path Select Switch --- This flag simply maps to the fault selected with the 'switchFaultId' configuration variable.
- Protection --- The protection switch raises (and latches) this fault flag if a switch over occurs. This fault maps to the 'switched' LED at the user interface. The flag is reset if you apply a 'RESET' command to the device (this command does nothing else for the Protection-Switch-1-To-1 device. Enable this fault if you want the switch over to be signaled as a latched fault.
- Invalid location --- This fault becomes active when the location detection neither reports the "side 1" location nor the "side 2" location. In a configuration with two antennas this

will probably mean that not all switches are set properly to route all signals via antenna 1 or antenna 2.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.operationMode	CHOICE	SAVE SETUP	EnumRange (SW-ALWAYS SW-ONCE)
config.switchMode	CHOICE	SAVE SETUP	EnumRange (SINGLE DUAL)
config.fault11Id	TEXT	SAVE SETUP	StringRange
config.fault12Id	TEXT	SAVE SETUP	StringRange
config.fault13Id	TEXT	SAVE SETUP	StringRange
config.fault14Id	TEXT	SAVE SETUP	StringRange
config.fault15Id	TEXT	SAVE SETUP	StringRange
config.fault21Id	TEXT	SAVE SETUP	StringRange
config.fault22Id	TEXT	SAVE SETUP	StringRange
config.fault23Id	TEXT	SAVE SETUP	StringRange
config.fault24Id	TEXT	SAVE SETUP	StringRange
config.fault25Id	TEXT	SAVE SETUP	StringRange
config.switchFaultId	TEXT	SAVE SETUP	StringRange

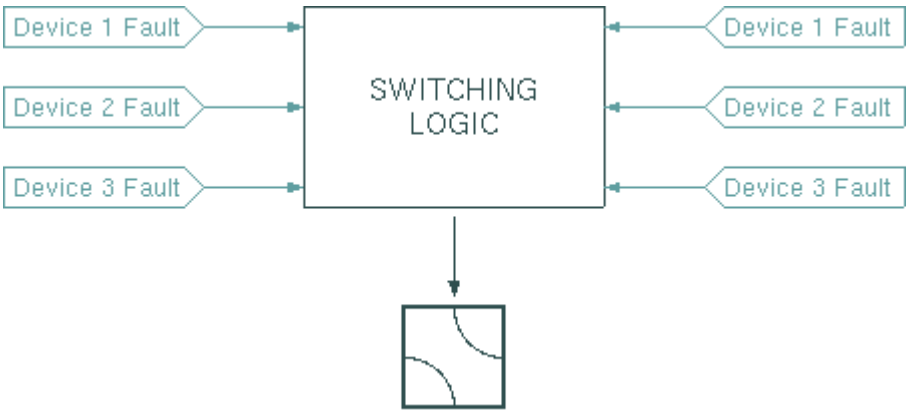
name	*type*	*flags*	*range*
config.switchId	TEXT	SAVE SETUP	StringRange
config.location1Val	TEXT	SAVE SETUP	StringRange
config.location2Val	TEXT	SAVE SETUP	StringRange
config.locdetectId	TEXT	SAVE SETUP	StringRange
config.locdetect1Val	TEXT	SAVE SETUP	StringRange
config.locdetect2Val	TEXT	SAVE SETUP	StringRange
config.suppressFaultIfOff	CHOICE	SAVE SETUP	EnumRange (YES NO)
location	CHOICE		EnumRange (1 2 INVALID)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
recovDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
reset	TEXT		StringRange
commanded	TEXT	R/O	StringRange R/O
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
singleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
doubleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Single fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Double fault)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Path select switch)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Invalid location)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switched)

9.535 Protection-Switch-1-To-1

The Protection-Switch-1-To-1 logical device provides a completely software based solution for a 1:1 redundancy switch. It saves the active chain (variable 'location') while the M&C is switched of and restores it at program start. This may cause a switchover at program start. For this reason, the Protection-Switch-1-To-1 logical device should only be used in applications where no position readback from the chain selecting switch is available. The Protection-Switch-NoSave-1-To-1 device provides the same functionality but relies in the switch position rather than saving the value.

The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch the (usually transmit-) chain.



As illustrated by the figure above, the redundancy switch reads up to four device fault variables for each chain and controls one switch. The latter not necessarily needs to be a waveguide switch as shown here. Any device may be configured to receive a command when the protection switch toggles the chain.

To control the switching behavior, the Protection-Switch-1-To-1 knows two configurable delay constants called 'switchDelay' and 'recovDelay'. Redundancy switch over takes place if these conditions become true at one time:

1. There was no switch over for at least 'recovDelay' milliseconds.
2. The opposite (actually redundant) chain states no fault.
3. There is a fault in at least one of the devices monitored for this (actually transmitting) chain for at least 'switchDelay' milliseconds.

When specifying the delay times, you should consider that the values become effective in steps of the IDLE time defined for the 'null' interface in the device setup. If the default value of 1000 msecs applies, all delays will be rounded up the next full second.

Redundancy switching is done fully symmetrically. There is no preferential chain. Hence the switch will **not** return to the initial position if the fault which caused the switch over disappears.

Operation Modes

As of version 2.02 / 080604 the Protection-Switch-1-To-1 device may be configured to act either in SW-ALWAYS or in SW-ONCE (single shot) mode.

SW-ALWAYS ist the default mode also used by earlier versions of the device. Like described above, the device will switch back to the first chain if the fault on this chain disappeared and the second chain shows a fault (eventually delayed by the recovDelay setting).

The SW-ONCE mode disables any further redundancy switching if the "switched" flag is set. This means the protection switch switches to the redundant chain and then inhibits itself. You have to click the "Reset" button at the device window to re-activate the device after a switchover.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. * --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
operationMode	may be set to SW-ONCE or SW-ALWAYS to control how the device handles consecutive faults. See the paragraph "Operation Modes" above for an explanation of this setting.
fault11Id	The message ID for the first fault flag to be monitored for chain 1. The the switch recognized device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well. Be aware that the latter are individually defined for each device driver.

parameter	*description*
fault12Id	The message ID for the second fault flag to be monitored for chain 1. You may leave this field empty if there is no second fault to monitor.
fault13Id	The message ID for the third fault flag to be monitored for chain 1. You may leave this field empty if there is no third fault to monitor.
fault14Id	The message ID for the fourth fault flag to be monitored for chain 1. You may leave this field empty if there is no fourth fault to monitor.
fault21Id	The message ID for the first fault flag to be monitored for chain 2.
fault22Id	The message ID for the second fault flag to be monitored for chain 2.
fault23Id	The message ID for the third fault flag to be monitored for chain 2.
fault24Id	The message ID for the fourth fault flag to be monitored for chain 2.
switchFaultId	The message ID of the fault flag indicating a problem with the chain select switch. Usually this is 'ZZZ.fault', if 'ZZZ' is the device name of the switch.
switchId	The message ID to send the switch over command to. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.
location1Val	The command to be sent to 'switchId' if the protection switch wants to select chain 1. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
location2Val	The command to be sent to 'switchId' if the protection switch wants to select chain 2. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
inhibitId	The message ID of a parameter to watch if the protection shall be inhibited.
inhibitVal	The value to compare the parameter to which is watched through the 'inhibitId' setting. Protection switching is inhibited if the value of the parameter addressed by 'inhibitId' matches the value of 'inhibitVal'. If you don't want to use this feature, leave both fields empty.
suppressFaultIfOff	may set to YES or NO. If set to YES, single fault and double fault are suppressed if protection is set to OFF. If set to NO, these faults are reported even if protection is OFF.

Fault definitions

The Protection-Switch-1-To-1 device defines a couple of individual faults to report the device's actual state. The fault conditions *always* are shown as LEDs at the first page if the device window. Depending on the individual fault configuration, these conditions also may be logged and summarized to a device fault for the redundancy switch. The table below describes the fault flags defined for the Protection-Switch-1-To-1 device.

- Single Fault --- Becomes true if at least one of the configured fault inputs becomes true. This also means that -- if the switch over already has occurred -- the protection switch actually is not able to compensate a possible fault in the remaining chain. Enable this fault if you want to be warned in this case.
- Double Fault --- In each chain at least one device reports a fault. The (transmit-) facility probably is not operable in this case.
- Path Select Switch --- This flag simply maps to the fault selected with the 'switchFaultId' configuration variable.
- Protection --- The protection switch raises (and latches) this fault flag if a switch over occurs. This fault maps to the 'switched' LED at the user interface. The flag is reset if you apply a 'RESET' command to the device (this command does nothing else for the Protection-Switch-1-To-1 device. Enable this fault if you want the switch over to be signaled as a latched fault.

Variables defined by this device driver

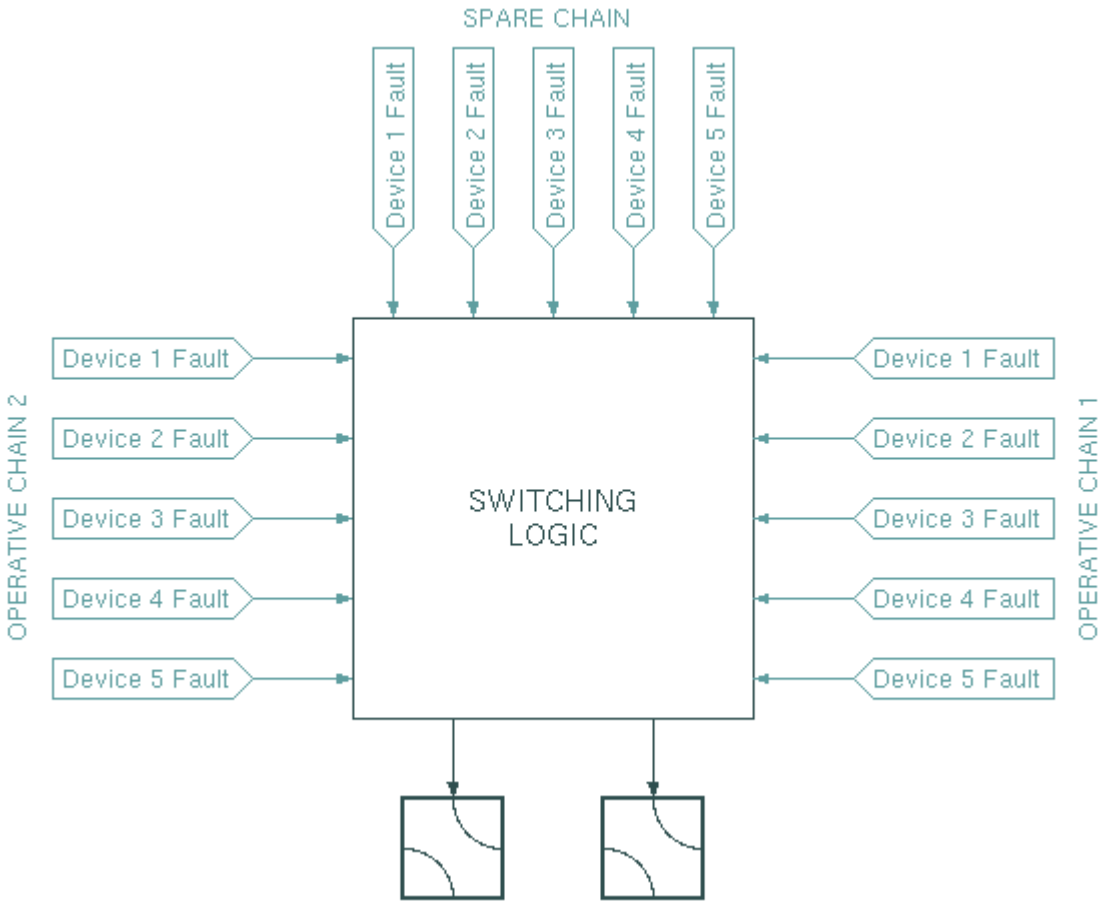
name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.operationMode	CHOICE	SAVE SETUP	EnumRange (SW-ALWAYS SW-ONCE)
config.fault11Id	TEXT	SAVE SETUP	StringRange
config.fault12Id	TEXT	SAVE SETUP	StringRange
config.fault13Id	TEXT	SAVE SETUP	StringRange
config.fault14Id	TEXT	SAVE SETUP	StringRange
config.fault15Id	TEXT	SAVE SETUP	StringRange
config.fault21Id	TEXT	SAVE SETUP	StringRange
config.fault22Id	TEXT	SAVE SETUP	StringRange
config.fault23Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.fault24Id	TEXT	SAVE SETUP	StringRange
config.fault25Id	TEXT	SAVE SETUP	StringRange
config.switchFaultId	TEXT	SAVE SETUP	StringRange
config.switchId	TEXT	SAVE SETUP	StringRange
config.location1Val	TEXT	SAVE SETUP	StringRange
config.location2Val	TEXT	SAVE SETUP	StringRange
config.inhibitId	TEXT	SAVE SETUP	StringRange
config.inhibitVal	TEXT	SAVE SETUP	StringRange
config.suppressFaultIfOff	CHOICE	SAVE SETUP	EnumRange (YES NO)
location	CHOICE	SAVE	EnumRange (1 2)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
recovDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
reset	TEXT		StringRange
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
singleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
doubleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Single fault)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Double fault)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Path select switch)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switched)

9.536 Protection-Switch-2-To-1

The Protection-Switch-2-To-1 logical device provides a completely software based solution for a 2:1 redundancy switch. The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch the (usually transmit-) chain.



As illustrated by the figure above, the redundancy switch reads up to five device fault variables for each chain and controls two switches. The latter not necessarily need to be waveguide switches as shown here. Any device may be configured to receive a command when the protection switch toggles the chain.

With a 2:1 redundancy, there is a dedicated spare chain which is placed into operation when

one of the operative chains fails. A redundancy switch over takes place if these conditions become true at one time:

1. The spare chain states no fault.
2. There is a fault in at least one of the devices monitored for this chain for at least 'switchDelay' seconds.

When specifying the delay times, you should consider that the values become effective in steps of the IDLE time defined for the 'null' interface in the device setup. If the default value of 1000 msecs applies, all delays will be rounded up the next full second even if you specify fractions of a second.

The Protection-Switch-2-To-1 also is able to switch back to the original equipment in case of a a failure in the spare chain. Such a 'reverse redundancy switching' is performed under the following conditions:

1. The original chain is completely OK for at least 'recovDelay' seconds.
2. There is a fault in at least one of the devices monitored for the spare chain for at least 'switchDelay' seconds.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
fault11Id	The message ID for the first fault flag to be monitored for the operative chain 1. The the switch recognized device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well. Be aware that the latter are individually defined for each device driver.
fault12Id ... fault15Id	The message IDs for up to four additional fault flags to be monitored for the operative chain 1. You may leave this fields empty if there are no more faults to monitor.

parameter	*description*
fault21Id ... fault25Id	The message IDs for the fault flags to be monitored for the operative chain 2.
faultS1Id ... faultS5Id	The message IDs for the fault flags to be monitored for the spare chain
switch1Id	The message ID for the first switch to control. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.
switch2Id	The message ID for the second switch to control. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.
location1Val	The commands to be sent to the switches if the protection switch wants to switch the spare equipment in place of chain 1. You have to enter the switch positions of S1 and S2, separated by a comma. Example: 'OFF,ON'
location2Val	The commands to be sent to the switches if the protection switch wants to switch the spare equipment in place of chain 2. You have to enter the switch positions of S1 and S2, separated by a comma. Example: 'OFF,ON'
locationSVal	The commands to be sent to the switches if the protection switch wants to switch the spare equipment into spare position. You have to enter the switch positions of S1 and S2, separated by a comma. Example: 'OFF,ON'
inhibitId	You may provide the message ID of a parameter here which the protection switch monitors and inhibits it's operation if the parameter reads a certain value.
inhibitVal	Enter the value here, the 'inhibitId' parameter is checked against. If the parameter matches this value, not redundancy switching takes place.

- Chain 1 --- A fault in one of the devices of Chain 1 is pending.
- Chain 2 --- A fault in one of the devices of Chain 2 is pending
- Spare --- A fault in one of the devices of Spare Chain is pending.
- Protection --- The protection switch raises (and latches) this fault flag if a switch over occurs. This fault maps to the 'switched' LED at the user interface. The flag is reset if you apply a 'RESET' command to the device (this command does nothing else for the Protection-Switch-1-To-1 device). Enable this fault if you want the switch over to be signaled as a latched fault.
- Switch configuration --- The switches are in an unknown or wrong position.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.fault11Id	TEXT	SAVE SETUP	StringRange
config.fault12Id	TEXT	SAVE SETUP	StringRange
config.fault13Id	TEXT	SAVE SETUP	StringRange
config.fault14Id	TEXT	SAVE SETUP	StringRange
config.fault15Id	TEXT	SAVE SETUP	StringRange
config.fault21Id	TEXT	SAVE SETUP	StringRange
config.fault22Id	TEXT	SAVE SETUP	StringRange
config.fault23Id	TEXT	SAVE SETUP	StringRange
config.fault24Id	TEXT	SAVE SETUP	StringRange
config.fault25Id	TEXT	SAVE SETUP	StringRange
config.faultS1Id	TEXT	SAVE SETUP	StringRange
config.faultS2Id	TEXT	SAVE SETUP	StringRange
config.faultS3Id	TEXT	SAVE SETUP	StringRange
config.faultS4Id	TEXT	SAVE SETUP	StringRange
config.faultS5Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.switch1Id	TEXT	SAVE SETUP	StringRange
config.switch2Id	TEXT	SAVE SETUP	StringRange
config.location1Val	TEXT	SAVE SETUP	StringRange
config.location2Val	TEXT	SAVE SETUP	StringRange
config.locationSVal	TEXT	SAVE SETUP	StringRange
config.inhibitId	TEXT	SAVE SETUP	StringRange
config.inhibitVal	TEXT	SAVE SETUP	StringRange
location	CHOICE	SAVE	EnumRange (OFF 1 2)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
recovDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
reset	TEXT		StringRange
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chainSFault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Chain 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Chain 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Spare)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch configuration)

9.537 Protection-Switch-Asymmetric

The Protection-Switch-Asymmetric logical device provides a software based solution for a 1:1 redundancy switch. As the name suggests, the device is asymmetric, it differentiates between a primary and a backup signal chain. In simple terms, the device switches to backup when there is a fault at the primary chain and it switches back to primary when the primary chain becomes healthy again.

In practice, things are a bit more sophisticated. Beside the primary chain fault there are other conditions which are to be taken into account, also there are a number of delay parameters which make the protection switch behave in a useful way.

Fault monitoring

The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch over. Actually the device supports five fault flags for each chain which are logically ORed to obtain the chain fault.

Inhibit signals

The device never switches to a chain which states a fault. Beside this, the device can monitor an additional fault flag for each chain which is interpreted as an inhibit signal. If this flag is true, the device will not switch to this chain, even if the opposite chain states a fault.

Delay pararameters

The Protection-Switch-Asymmetric logical device provides three delay parameters which control the device's behaviour in different situations.

- Switchdelay --- This delay controls the time, a fault condition must be present before the device switches from primary to backup
- Recoverydelay --- If the device switches to backup, the primary chain states OK again and a fault happens at the backup chain, the device switches back to the primary chain after this time. This delay prevents the redundancy switch from oscillating if the switch over process itself causes a short fault at the backup chain.
- Switchbackdelay --- This delay controls the time, after that the device switches back to primary when no more faults are active. This time usually is set to a much longer value than the recovery delay.

All delay values may be set to zero resulting in an immediate reaction of the device.

Switchover conditions

With protection enabled, the device switches from primary to backup if the backup chain shows no fault and no inhibit condition, but the primary chain signals a fault and this condition stays true for at least 'switch delay' seconds.

The device switches back from backup to primary if the primary chain shows no fault and no inhibit condition, but the backup chain signals a fault and this condition stays true for at least 'recovery delay' seconds.

The device also switches back from backup to primary if the primary chain shows no fault and

no inhibit condition, regardless of the state of the backup chain with this condition being true for at least 'switch back delay' seconds.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. * --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
faultPri1Id ... faultPri5Id	The message IDs for the fault flags to be monitored for the primary chain. The device recognizes device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well.
inhibitPriId	The message ID of a fault flag to watch if the protection shall be inhibited switching back to primary.
faultBak1Id ... faultBak5Id	The message IDs for the fault flags to be monitored for the backup chain. The device recognizes device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well.
inhibitBakId	The message ID of a fault flag to watch if the protection shall be inhibited switching to backup.
switchFaultId	The message ID of the fault flag indicating a problem with the chain select switch. Usually this is 'ZZZ.fault', if 'ZZZ' is the device name of the switch.
switchId	The message ID to send the switch over command to. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.
switchPriVal	The command to be sent to 'switchId' if the protection switch wants to select the primary chain. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
switchBakVal	The command to be sent to 'switchId' if the protection switch wants to select the backup chain. Usually this is one of 'ON', 'OFF', 'A' or 'B'.

parameter	*description*
locdetectId	The message ID to monitor in order to detect the actual primary/backup state. This may be (in case of a simple switch) the same as switchId, but if switchId for example triggers a macro to perform a more complex switchover procedure you may monitor another parameter to detect if the primary or the backup chain is selected.
locdetectPriVal	The value expected from the parameter addressed by locdetectId for the primary chain
locdetectBakVal	The value expected from the parameter addressed by locdetectId for the backup chain

Variables defined by this device driver

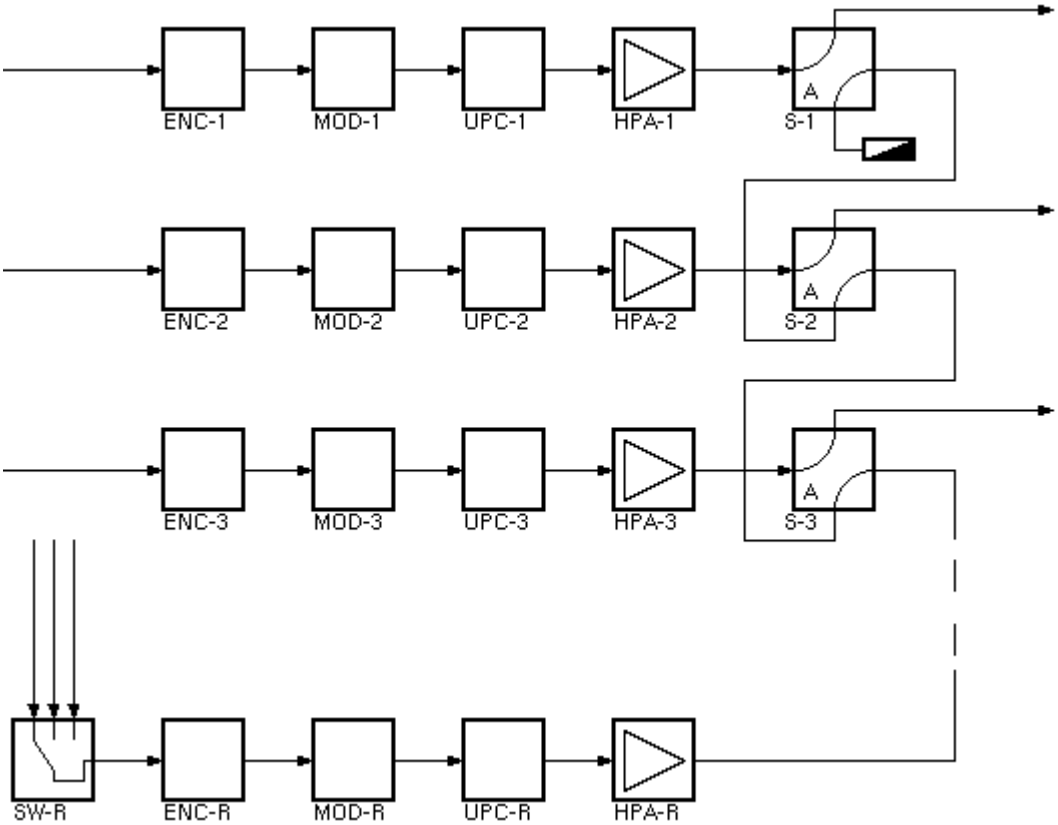
name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.faultPri1Id	TEXT	SAVE SETUP	StringRange
config.faultPri2Id	TEXT	SAVE SETUP	StringRange
config.faultPri3Id	TEXT	SAVE SETUP	StringRange
config.faultPri4Id	TEXT	SAVE SETUP	StringRange
config.faultPri5Id	TEXT	SAVE SETUP	StringRange
config.inhibitPriId	TEXT	SAVE SETUP	StringRange
config.faultBak1Id	TEXT	SAVE SETUP	StringRange
config.faultBak2Id	TEXT	SAVE SETUP	StringRange
config.faultBak3Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.faultBak4Id	TEXT	SAVE SETUP	StringRange
config.faultBak5Id	TEXT	SAVE SETUP	StringRange
config.inhibitBakId	TEXT	SAVE SETUP	StringRange
config.switchFaultId	TEXT	SAVE SETUP	StringRange
config.switchId	TEXT	SAVE SETUP	StringRange
config.switchPriVal	TEXT	SAVE SETUP	StringRange
config.switchBakVal	TEXT	SAVE SETUP	StringRange
config.locdetectId	TEXT	SAVE SETUP	StringRange
config.locdetectPriVal	TEXT	SAVE SETUP	StringRange
config.locdetectBakVal	TEXT	SAVE SETUP	StringRange
location	CHOICE		EnumRange (PRIMARY BACKUP)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
recovDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
switchBackDelay	FLOAT	SAVE	DoubleRange (0.0 .. 3600.0)
reset	TEXT	NOPRESET	StringRange
sbReload	TEXT	NOPRESET	StringRange
sbNow	TEXT	NOPRESET	StringRange
primaryFault	BOOLEAN	R/O	BooleanRange R/O (true false)
backupFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
switchBackCountDown	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Primary fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Backup fault)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Path select switch)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.538 Protection-Switch-N-To-1

The Protection-Switch-N-To-1 logical device provides a completely software based solution for a N:1 redundancy switch. The device is preconfigured to handle 2 to 12 operative chains, but it may quite simply be extended to serve more chains. The block diagram below shows a typical application scenario for the Protection-Switch-N-To-1 device. Three operative transmit chains are backed up by one spare chain. A 'ladder' of waveguide switches at the chains' output controls which RF signal gets routed to the antenna.



During normal operation the waveguide switches are set as shown in the diagram. The output of the operative chains goes to the antenna, the output of the spare chain is routed to a dummy load. The protection switch watches (mirrors in memory) the key settings of all devices in the operative chains. In case of a fault in an operative chain, the spare chain gets gets switched in place of the faults chain and the spare devices are commanded to the settings which have been mirrored for the faulty chain.

The protection switch operates in 'single shot mode'. It switches once if a fault was detected, but never switches back automatically. After a switch-over to the spare chain, the operator has to check the equipment in the faulty chain. After everything is OK again, the signal can be switched back to the operative chain at a suitable time. Only now the transmission is backed up again.

The behaviour of the Protection-Switch-N-To-1 device is widely configurable, however to keep the number of configuration settings manageable, the device demands some rules in the overall structure and the device naming. These rules are:

- The devices which are managed by the protection switch must follow certain naming conventions. In the protection switch device setup dialog name prefixes for the device in each chain can be defined, the protection switch completes these names by the chain index on it's own. Alternatively the device names may be defined explicitly as comma separated lists.
- If using waveguide switches the the chain outputs, the switches must be connected to a ladder as shown in the block diagram above. All switches in the ladder must be oriented in the same way. The naming rules apply for the switches as well.
- The RF output power of each chain must be controlled by either aLevel-Set , a Level-Control or a Delay-Line logical device (not shown in the block diagram).

Fault monitoring

The protection switch monitors the faults in each chain and controls the switch-over from these faults. A fault in an operative chain causes the protection switch to replace the faulty chain by the spare one. In contrast, a fault in the redundant chain prevents the protection switch from using the spare chain.

The device provides two modes of fault monitoring. If the 'faultMode' configuration parameter ist set to 'ALL FAULTS', the device monitor the summary fault states of all devices in a chain. If at least one faulty device is recognized and pending for 'switchDelay' seconds, a switch-over is triggered.

The device summary faults normally comprise all fault flags of a device including the communication alarm. There may be fault conditions which do not affect the RF performance of a chain and therefore should not trigger a switch-over. You may set such fault condition to the fault priority 'INFO' at each individual device to prevent them from being summarized in the device fault state.

The fault mode setting "HW FAULTS ONLY" ignores all device faults in the chains except for the fault stated by the 'hwFaultPrefix' devices. This mode is used to let the protection switch make up it's switch-over decision only from a fault reported by a relay circuit. Multiple relay circuits

may be combined by hardware wiring or by sat-nms logical devices like the Gate-Array , Logical-Gate or the Logical-Gate-3 devices.

Klystron support

The protection switch supports the usage of a klystron amplifier in the spare chain. The klystron mode is enabled by defining a list of channel selector settings in the configuration / setup dialog.

In klystron mode, the protection switch commands a channel selector setting to the klystron amplifier during the switch-over process. RF power is kept OFF during the switchover until the klystron device states that it has successfully tuned to the commanded channel.

Switch-over and mirroring

If the device detects a fault in one of the chains and protection is enabled (globally for the device and individually for this chain), the device switches the redundant chain in place of the faulty one. The following happens:

1. In case of a klystron amplifier in the redundant chain, the channel selector setting for the chain to replace is applied to the klystron amplifier.
2. The input switch in front of the redundant chain is set to route the input signal of the defective chain to the redundant one.
3. If a "pre mirror delay" has been configured, the switchover procedure waits this time.
4. The parameters for the upconverter, modulator and auxiliary devices are mirrored from the defective to the redundant chain.
5. The level calibration values are copied as well
6. If a "post mirror delay" has been configured, the switchover procedure waits this time.
7. With a klystron amplifier, the switchover procedure waits until the amplifier's channel adjustment has completed. There is a 120 seconds timeout for this, the switchover gets aborted and a fault ist raised if the amplifier does not complete within this time.
8. If configured, the procedure checks the redundant chain input level and waits until this reaches a configurable threshold. For this a timeout value of 5 seconds applies, if the input level at the redundant chain is too low after this time the procedure aborts with a fault flag raised.
9. Now the EIRP setting is applied to the redundant chain to set the amplifier to it's nominal power. If an puling power control is configured and active, the device mirrors the UPC nominal output level instead. It applies a level offset to compensate the signal path addentuation, the offset is defined in the UPC device.
10. The output switches are set to route the defective chain to dummy load and the redundant chain to the antenna.

If the protection switch device is configured to the switch mode "RF-ON" instead of "WAVEGUIDE", step 9 is done slightly different: No switch is actuated, but the redundant EIRP controller is commanded RF-ON while the controller in the defective chain is commanded RF-OFF.

Switch faults

The protection switch device also monitors the fault state of the output switches and the fault state of the input selection switch of the redundant chain as well. If the device detects a fault at one of the switches, it raises a "Switch hardware" fault. This situation blocks any redundancy switching.

Temperature fault monitoring

The protection switch is capable to monitor an additional fault signal for each chain which reports a temperature alarm caused by the RF power generated in this chain. Dummy loads or filter/power combiners are possible sources for such a temperature fault. If such a fault is detected, the protection switch commands the faulty chain RF-OFF and disables protection switching for this chain.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
switchMode	If set to 'WAVEGUIDE', the device will command a chain of waveguide switches in order to switch the spare chain in place of one of the operative chains. If set to 'RF-ON', the device will control the RF-ON settings of the EIRP control devices in each chain for this task.
faultMode	If set to 'ALL FAULTS', the summary faults of all devices in a chain will trigger a switch-over. Both, "WARNING" and "Summary FAULT" conditions will trigger a switch-over in this case. If set to 'HW FAULT ONLY' the device will monitor only the fault messages sent by the device identified by 'hwFaultPrefix' + chain-index.
firstIndex	The number appended to the name prefixes for the first operative chain (usually 1). If both, firstIndex and lastIndex are set to 0 the device does not apply any automatic indexing but interprets the *???Prefix* settings as comma separated lists of device names.

parameter	***description***
lastIndex	The number appended to the name prefixes for the last operative chain (usually the number of chains to backup). If both, firstIndex and lastIndex are set to 0 the device does not apply any automatic indexing but interprets the *???Prefix* settings as comma separated lists of device names.
hpaPrefix	The name prefix for the HPA device in each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
upconverterPrefix	The name prefix for the upconverter device in each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
modulatorPrefix	The name prefix for the modulator device in each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
levelPrefix	The name prefix for the Level-Set / Level-Control device in each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
auxPrefix	The name prefix for an auxiliary (e.g. MPEG Encoder) device in each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
aux2Prefix	Like auxPrefix for another set of devices.
hwFaultPrefix	The name prefix for the a device in each chain which monitors the fault relay circuits of the equipment in the chain. Logical OR-ing of multiple devices may be done by hardware wiring or by logical devices provided by the sat-nms software. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.

parameter	***description***
tempFaultPrefix	The name prefix for the a device in each chain which monitors a temperature alarm caused by the RF power generated in this chain. A fault recognized by this device will switch the chain to RF-OFF and disabled protection for this chain. A logical combination of multiple faults may be done by hardware wiring or by logical devices provided by the sat-nms software. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
sw1Device	The device name of the N:1 input switch in front of the spare chain. '.position' is automatically appended. You also may specify the complete message id of the switch position variable, e.g. 'MATRIX.output.4'.
sw1Positions	The switch position to be applied to the input switch. This is a comma separated list of switch settings. The first value designates the the switch setting in for first chain, the second one for the second chain and so on. There is no setting for the spare chain as the protection switch by default leaves this switch unchanged when switching back to 'NONE'.
sw1NonePosition	The switch position to be applied to the input switch when switching back to NONE. If empty, the the input switch remains at its least position, no position is commanded in this case.
sw2Prefix	The name prefix for the waveguide switches at the end of each chain. The real device names are built by the device by appending the chain index to this string. If indexing is disabled, this is a comma separated list of device names.
sw2NrmCmd	The waveguide switch setting that switches a chain to the antenna.
sw2BackupCmd	The waveguide switch setting that switches the spare chain in place of this chain.
klystronChSelect	The channel selector settings to be set at a klystron amplifier in the spare position. This is a comma separated list of settings like the 'sw1Positions' list. See the paragraph 'Klystron support' above for details. Leave this field empty if the spare HPA is not a klystron amplifier.
levelSetType	The type of level set devices the protection switch shall control. When set to LEVEL-CONTROL, the device expects Level-Control or Level-Set devices. When set to DELAY-LINE, the device expects Delay-Line devices instead.

parameter	***description***
preMirrorDelay	If this time (milliseconds) is set to a non-zero value, the switchover delays at least this time after switching the redundant chain input but before mirroring the chain parameters. The true delay time is rounded up to the next multiple of the device driver cycle (IDLE) time.
postMirrorDelay	If this time (milliseconds) is set to a non-zero value, the switchover delays at least this time after mirroring the chain parameters but before applying the EIRP setting. The true delay time is rounded up to the next multiple of the device driver cycle (IDLE) time.
redInputLevelId	If this field is not empty, the device treats it as the message identifier for the redundant chain input level. The value is checked against the threshold defined below during the switchover procedure. The procedure only proceeds to set the EIRP, if this level is above the threshold. If redInputLevelId is empty, no level check is done.
redInputLevelThresh	The threshold value for the level check described above.
abortOnSpareFault	If this is set ON, the device will check the spare chain fault just before the switches are actuated to switch the spare chain in place. If a spare chain fault has been raised during the mirroring procedure before, the switchover will be aborted and a fault is set.
standbyUpcOutput	If this parameter is not empty, it is interpreted as a UPC nominal output level which is set ar the redundant chain when the protection is switched back to NONE. Only applicable if UPC devices are configured.
userInterface	DROP-DOWN makes the device window to use a drop down control to select the chain to be backed up (the default user interface), BUTTONS selects a device screen using a sett of buttons to do the selection. Changes to this setting become visible when the device window gets opened the next time.

Changes to most of the configuration parameter do not become effective until the VLC software is re-started. This is due to the fact, that the protection switch software has to allocate sat-nms parameters dynamically, depending on the number of chains configured. To signal this, the software sets the 'Config mismatch' fault flag as soon as such a configuration parameter is changed.

Fault descriptions

- Spare chain --- At least one device in the redundant chain states a fault.
- Switch configuration --- The switches are in an unknown or wrong position.
- Switch hardware --- At least one of the signal switches controlled by this device states a

fault.

- Config mismatch --- This fault flag is set if changed or incorrect configuration settings prevent the device from normal operation. The 'state' variable at the 'info' page describes the reason of this fault:
  - RESTARTREQUIRED! --- Changes to most of the device's configuration parameters do not become effective until the VLC server software is re-started. If you change such a parameter, the 'Config mismatch' fault is set with this description. This and further changes to configuration parameters are stored properly, you can set all configuration parameter if the device in one pass and then restart the software.
  - NOCHAINS --- No chains were detected to configure. Probably you configured the device in RF-ON switchmode with comma separated device name lists. A dummy list of WG switches must be entered in this case (see below).
  - BADINDEXORDER --- You must define a 'lastIndex' > 'firstIndex'
  - TOOMUCHCHAINS --- The number of chains must not exceed the limit defined in the device driver file. Actually this limit is 12. In the device driver file you find a description how to extend this limit.
  - BADINDEX --- The index numbers must not exceed 99.
  - HWFAULTMISMATCH --- You've set the device to 'HW FAULT ONLY' mode, but did not specify a 'hwFaultPrefix'. The device will never detect a fault in this configuration and will never initiate a switch-over.
  - BADINPUTSWITCHTABLE --- The 'sw1Positions' configuration parameter does not device enough switch positions for the number of chains you defined. 'sw1Positions' must contain an input switch setting for each chain and additionally the switch setting for the setting 'NONE'
- Klystron channel --- Channel adjustment at the spare klystron amplifier timed out. 120 seconds after commanding the channel the amplifier still reported "adjust in progress". The switchover process has been aborted, the Protection-Switch-N-To-1 device is locked. **This is a latching fault, you have to click RESET at the maintenance page to reset/release the fault condition**
- Spare input level --- A spare chain input level check has been configured and the level did not reach the threshold value within 5 seconds. The switchover process has been aborted, the Protection-Switch-N-To-1 device is locked. **This is a latching fault, you have to click RESET at the maintenance page to reset/release the fault condition**
- Protect enable --- The global protection enable switch is OFF. By default this condition is treated as a fault. You may change the priority setting of this fault to "NONE" if you want to change thus behaviour.
- Chain-? protectenable --- The protection enable switch for this particular chain is OFF. By default this condition is treated as a fault. You may change the priority setting of this fault to "NONE" if you want to change thus behaviour. **This is a latching fault, the next switchover will reset it. Alternatively you may click RESET at the maintenance page to reset/release the fault condition**

Parameter mirroring

The table below states which parameters are remembered and set at the devices of the spare chain in case of a switch-over.

device	***name config setting***	***mirrored parameters***
HPA	hpaPrefix	No settings are mirrored. Changing the HPA settings would interfere with the level control device. In klystron amplifier mode, the channel selector setting is applied from a list defined in the protection switch device setup.
UPCONVERTER	upconverterPrefix	The transmit frequency setting is mirrored.
MODULATOR	modulatorPrefix	All configurable parameters which are not setup parameters are mirrored. Parameters marked with NOPRESET are excluded from mirroring as well.
EIRP control	levelPrefix	The EIRP value and calibration file name configuration setting are mirrored.
AUXILIARY (e.g. encoder)	auxPrefix, aux2Prefix	All configurable parameters which are not setup parameters are mirrored. Parameters marked with NOPRESET are excluded from mirroring as well.

Standard and non-standard device indexing

The Protection-Switch-N-To-1 device provides an easy method to define the names of the devices referenced by the protection switch. The names are defined as prefixes, index are appended automatically to get the complete names. If e.g. firstIndex is 1 and lastIndex is 2, a prefix value "HPA-" would expand to the device names "HPA-R" for the spare device and "HPA-1", "HPA-2" for the operative devices at chains 1, 2.

There are situations where this naming scheme cannot be applied. This may happen if the device naming must match external conventions of if multiple protection switches are configured in one system, automatically indexed names would conflict. Specially the waveguide switch names often are critical for this. In such a situation the device names may be defined explicitly at the Protection-Switch-N-To-1 setup as described below:

1. Set firstIndex and lastIndex to 0
2. With sw2Prefix define the names of the waveguide switches at the end of each chain as a comma separated list of device names. The number of switch names defines the number of chains the device handles, the first name is assigned to chain 1. Example: "S22,S23" defines two operative chains, S22 routes the output of chain 1, S23 the output of chain 2.
3. For all devices you want to monitor or mirror, fill a comma separated list of device names into the *???Prefix* field in question. All these lists must consist of*n+1* device names, the first one for the chain 1 device name, the last one for the spare chain device name. Example: "ENC-13,ENC-21,ENC-R2" defines two operative devices ENC-13 (chain 1) and ENC-21 (chain 2), ENC-2R as the spare device for these.
4. A the sw2Prefix list determines the number of chains to operate, this list must be present even if the RF-ON switch mode is used. Enter a list of dummy device names in this case, one name for each chain to operate.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.chains	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.sequencing	TEXT	R/O	StringRange R/O
info.usesDelayLine	BOOLEAN	R/O	BooleanRange R/O (true false)
info.switching	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.switchMode	CHOICE	SAVE SETUP	EnumRange (WAVEGUIDE RF-ON)
config.faultMode	CHOICE	SAVE SETUP	EnumRange (ALL FAULTS HW FAULT ONLY)
config.firstIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lastIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hpaPrefix	TEXT	SAVE SETUP	StringRange
config.upconverterPrefix	TEXT	SAVE SETUP	StringRange
config.modulatorPrefix	TEXT	SAVE SETUP	StringRange
config.levelPrefix	TEXT	SAVE SETUP	StringRange
config.upcPrefix	TEXT	SAVE SETUP	StringRange
config.auxPrefix	TEXT	SAVE SETUP	StringRange
config.aux2Prefix	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.hwFaultPrefix	TEXT	SAVE SETUP	StringRange
config.tempFaultPrefix	TEXT	SAVE SETUP	StringRange
config.abortOnSpareFault	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.sw1Device	TEXT	SAVE SETUP	StringRange
config.sw1Positions	TEXT	SAVE SETUP	StringRange
config.sw1NonePosition	TEXT	SAVE SETUP	StringRange
config.sw2Prefix	TEXT	SAVE SETUP	StringRange
config.sw2NrmCmd	TEXT	SAVE SETUP	StringRange
config.sw2BackupCmd	TEXT	SAVE SETUP	StringRange
config.klystronChSelect	TEXT	SAVE SETUP	StringRange
config.levelSetType	CHOICE	SAVE SETUP	EnumRange (LEVEL-CONTROL DELAY-LINE)
config.preMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30000)
config.postMirrorDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30000)
config.redInputLevelId	TEXT	SAVE SETUP	StringRange
config.redInputLevelThresh	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.standbyUpcOutput	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.userInterface	CHOICE	SAVE SETUP	EnumRange (DROP-DOWN BUTTONS)
locationStored	TEXT	SAVE	StringRange
location	CHOICE		EnumRange (NONE)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
chain.0.name	TEXT	R/O	StringRange R/O
chain.1.name	TEXT	R/O	StringRange R/O
chain.2.name	TEXT	R/O	StringRange R/O
chain.3.name	TEXT	R/O	StringRange R/O
chain.4.name	TEXT	R/O	StringRange R/O
chain.5.name	TEXT	R/O	StringRange R/O
chain.6.name	TEXT	R/O	StringRange R/O
chain.7.name	TEXT	R/O	StringRange R/O
chain.8.name	TEXT	R/O	StringRange R/O
chain.9.name	TEXT	R/O	StringRange R/O
chain.10.name	TEXT	R/O	StringRange R/O
chain.11.name	TEXT	R/O	StringRange R/O
chain.12.name	TEXT	R/O	StringRange R/O
chain.1.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.2.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.3.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.4.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.5.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.6.enable	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
chain.7.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.8.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.9.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.10.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.11.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.12.enable	CHOICE	SAVE	EnumRange (OFF ON)
chain.1.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.9.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.10.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.11.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.12.protected	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
chain.9.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.10.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.11.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.12.replaced	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.0.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.9.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.10.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.11.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.12.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Spare chain)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch configuration)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch hardware)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Config mismatch)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Klystron channel)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Spare input level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protect enable)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Chain-1 protect enable)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Chain-2 protect enable)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Chain-3 protect enable)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Chain-4 protect enable)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Chain-5 protect enable)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Chain-6 protect enable)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Chain-7 protect enable)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Chain-8 protect enable)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Chain-9 protect enable)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Ch-10 protect enable)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Ch-11 protect enable)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Ch-12 protect enable)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switching aborted)

9.539 Protection-Switch-N-To-M

The Protection-Switch-N-To-M logical device builds together with a set of protection controller devices a n:m redundancy solution for satellite modulators, downconverters etc.

This solution supports a pool of backup devices from which one device is taken in case of failure of an operative device. In case of a redundancy switch all parameters are mirrored to the spare device.

Actually there are two typed of protection controller devices available:

- Modem-Protection-Controller for satellite modulators which fieed their signals into a crosspoint oriented switch matrix and select their input signal from a network stream.
- Downconverter-Protection-Controller for downconverters which get their input signals from a distributive switch matrix and send their output signals to another distributive matrix.

The devices managed by Protection-Switch-N-To-M device must be all of the same type, to make an 1:1 parmeter mirroring to the spare device work properly, the manufacturer and model type should also be the same

For n:m redundancy, one Protection-Switch-N-To-M device and for each managed modulator a protection controller device are configured in the M&C device setup. This distribution across two device types enables a very flexible configuration of the redundancy, so there is no explicit limit on the number of managed modulators.

The Protection-Switch-N-To-M device, as the central instance of the redundancy switch, manages the pool of modems (which are operational, which are spare). It defines the name of the switching matrix that routes the outputs of the modulators as well as parameters shared by the modem protection controller devices as configuration parameters.

The device's operational parameters are:

- Global protection ON/OFF
- Global status display in the form of a multi-line text field that summarizes the configuration and current status of all modulators involved.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there is one configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and

control.

parameter	*description*
matrix	The name of the switch matrix which routes the outputs of all managed modulators to the transmit equipment.

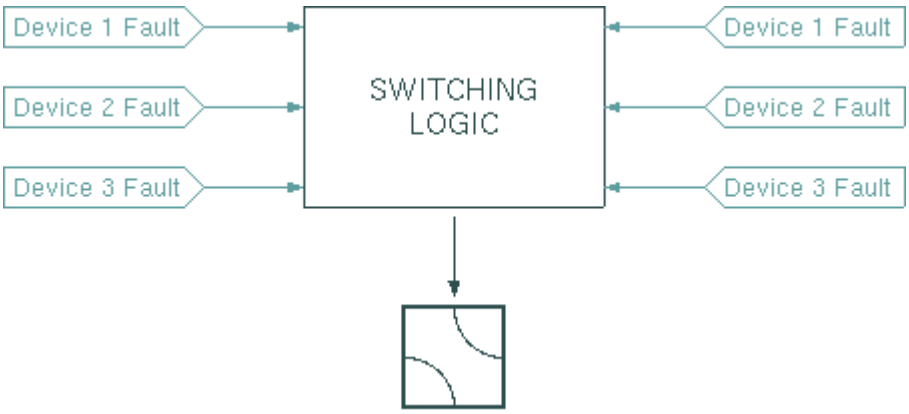
Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.freeSpares	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.matrix	TEXT	SAVE SETUP	StringRange
protection	CHOICE	SAVE	EnumRange (OFF ON)
state	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.03	ALARM	R/O	AlarmFlagRange R/O (No spare available)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Mixed controllers)

9.540 Protection-Switch-NoSave-1-To-1

The Protection-Switch-NoSave-1-To-1 logical device provides a completely software based solution for a 1:1 redundancy switch. It is almost the same as the Protection-Switch-1-To-1 , but it does not save the chain select (location) variable to disk like the latter one does.

The logical device monitors the fault states of a couple of devices, deriving from this the information whether to switch the (usually transmit-) chain.



As illustrated by the figure above, the redundancy switch reads up to four device fault variables for each chain and controls one switch. The latter not necessarily needs to be a waveguide switch as shown here. Any device may be configured to receive a command when the protection switch toggles the chain.

To control the switching behavior, the Protection-Switch-1-To-1 knows two configurable delay constants called 'switchDelay' and 'recovDelay'. Redundancy switch over takes place if these conditions become true at one time:

1. There was no switch over for at least 'recovDelay' milliseconds.
2. The opposite (actually redundant) chain states no fault.
3. There is a fault in at least one of the devices monitored for this (actually transmitting) chain for at least 'switchDelay' milliseconds.

When specifying the delay times, you should consider that the values become effective in steps of the IDLE time defined for the 'null' interface in the device setup. If the default value of 1000 msecs applies, all delays will be rounded up the next full second.

Redundancy switching is done fully symmetrically. There is no preferential chain. Hence the switch will **not** return to the initial position if the fault which caused the switch over disappears.

Operation Modes

The Protection-Switch-NoSave-1-To-1 device may be configured to act either in SW-ALWAYS or in SW-ONCE (single shot) mode.

SW-ALWAYS is the default mode also used by earlier versions of the device. Like described above, the device will switch back to the first chain if the fault on this chain disappeared and the second chain shows a fault (eventually delayed by the recovDelay setting).

The SW-ONCE mode disables any further redundancy switching if the "switched" flag is set. This means the protection switch switches to the redundant chain and then inhibits itself. You have to click the "Reset" button at the device window to re-activate the device after a switchover.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page. * --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the redundancy switch with the devices it shall monitor and control.

parameter	***description***
operationMode	may be set to SW-ONCE or SW-ALWAYS to control how the device handles consecutive faults. See the paragraph "Operation Modes" above for an explanation of this setting.
fault11Id	The message ID for the first fault flag to be monitored for chain 1. The the switch recognized device summary fault flags ('XXXXX.fault') and individual fault flags ('XXXXX.faults.yy') as well. Be aware that the latter are individually defined for each device driver.
fault12Id	The message ID for the second fault flag to be monitored for chain 1. You may leave this field empty if there is no second fault to monitor.
fault13Id	The message ID for the third fault flag to be monitored for chain 1. You may leave this field empty if there is no third fault to monitor.
fault14Id	The message ID for the fourth fault flag to be monitored for chain 1. You may leave this field empty if there is no fourth fault to monitor.
fault21Id	The message ID for the first fault flag to be monitored for chain 2.
fault22Id	The message ID for the second fault flag to be monitored for chain 2.
fault23Id	The message ID for the third fault flag to be monitored for chain 2.
fault24Id	The message ID for the fourth fault flag to be monitored for chain 2.
switchFaultId	The message ID of the fault flag indicating a problem with the chain select switch. Usually this is 'ZZZ.fault', if 'ZZZ' is the device name of the switch.
switchId	The message ID to send the switch over command to. Usually this is 'ZZZ. position' if 'ZZZ' is the name of the switch device.

parameter	*description*
location1Val	The command to be sent to 'switchId' if the protection switch wants to select chain 1. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
location2Val	The command to be sent to 'switchId' if the protection switch wants to select chain 2. Usually this is one of 'ON', 'OFF', 'A' or 'B'.
inhibitId	The message ID of a parameter to watch if the protection shall be inhibited.
inhibitVal	The value to compare the parameter to which is watched through the 'inhibitId' setting. Protection switching is inhibited if the value of the parameter addressed by 'inhibitId' matches the value of 'inhibitVal'. If you don't want to use this feature, leave both fields empty.

Fault definitions

The Protection-Switch-NoSave-1-To-1 device defines a couple of individual faults to report the device's actual state. The fault conditions *always* are shown as LEDs at the first page if the device window. Depending on the individual fault configuration, these conditions also may be logged and summarized to a device fault for the redundancy switch. The table below describes the fault flags defined for the Protection-Switch-1-To-1 device.

- Single Fault --- Becomes true if at least one of the configured fault inputs becomes true. This also means that -- if the switch over already has occurred -- the protection switch actually is not able to compensate a possible fault in the remaining chain. Enable this fault if you want to be warned in this case.
- Double Fault --- In each chain at least one device reports a fault. The (transmit-) facility probably is not operable in this case.
- Path Select Switch --- This flag simply maps to the fault selected with the 'switchFaultId' configuration variable.
- Protection --- The protection switch raises (and latches) this fault flag if a switch over occurs. This fault maps to the 'switched' LED at the user interface. The flag is reset if you apply a 'RESET' command to the device (this command does nothing else for the Protection-Switch-1-To-1 device. Enable this fault if you want the switch over to be signaled as a latched fault.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

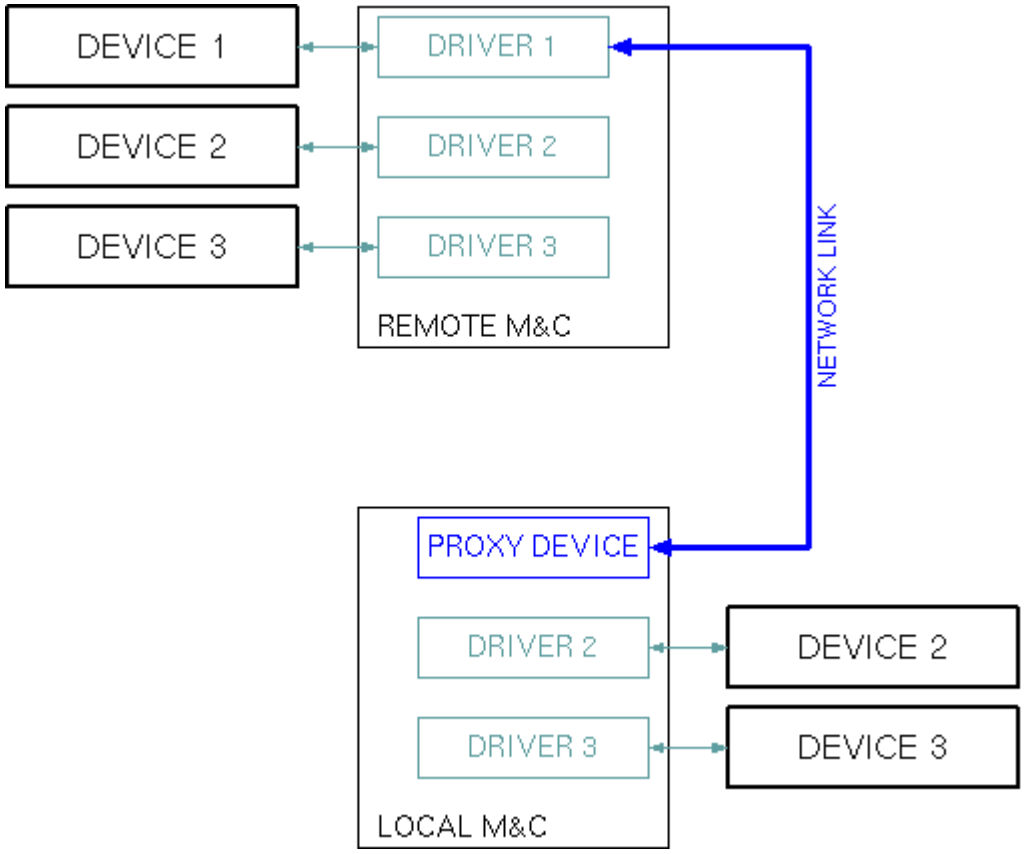
name	*type*	*flags*	*range*
config.operationMode	CHOICE	SAVE SETUP	EnumRange (SW-ALWAYS SW-ONCE)
config.fault11Id	TEXT	SAVE SETUP	StringRange
config.fault12Id	TEXT	SAVE SETUP	StringRange
config.fault13Id	TEXT	SAVE SETUP	StringRange
config.fault14Id	TEXT	SAVE SETUP	StringRange
config.fault15Id	TEXT	SAVE SETUP	StringRange
config.fault21Id	TEXT	SAVE SETUP	StringRange
config.fault22Id	TEXT	SAVE SETUP	StringRange
config.fault23Id	TEXT	SAVE SETUP	StringRange
config.fault24Id	TEXT	SAVE SETUP	StringRange
config.fault25Id	TEXT	SAVE SETUP	StringRange
config.switchFaultId	TEXT	SAVE SETUP	StringRange
config.switchId	TEXT	SAVE SETUP	StringRange
config.location1Val	TEXT	SAVE SETUP	StringRange
config.location2Val	TEXT	SAVE SETUP	StringRange
config.inhibitId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inhibitVal	TEXT	SAVE SETUP	StringRange
location	CHOICE		EnumRange (1 2)
protection	CHOICE	SAVE	EnumRange (OFF ON)
switchDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
recovDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
reset	TEXT		StringRange
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
singleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
doubleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Single fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Double fault)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Path select switch)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switched)

9.541 Proxy-Device

The Proxy-Device logical device maps all parameters of a device which is controlled by a remote sat-nms M&C into the local system. A typical application for the Proxy-Device is to couple M&C systems for site redundancy switching.

On startup or when the Proxy-Device receives a new 'address' configuration setting, it connects to the remote M&C and checks which type the remote device is. Then it sets up all variables, this device type defines. It binds these variables to the device remote M&C, to make each variable at the local system an exact duplicate of the variable at the remote system.



Once it is connected to the remote system, the Proxy-Device also changes it's appearance at the user interface: The device window pages now are those of the remote device. The faults page shows the faults of the remote device, the info page no longer describes the device driver as a Proxy-Device but as the remote device type it mirrors.

For all that the Proxy-Device does not resemble the remote device in all details. There are some few exceptions which are important to manage the Proxy-Device:

- The Proxy-Device does not mirror the configuration / setup variables of the remote device. The setup dialog which is accessible through the maintenance page of the device window always shows the Proxy-Device's setup variables, not the setup variables of the remote device. This ensures that the Proxy-Device remains (re-) configurable even it is already connected to the remote M&C.
- The Proxy-Device adds one fault flag ('Remote connection fault') to the list of fault flags the remote device provides. This fault is set while the IP link to the remote system is broken. The remote connection fault contributes to the summary fault in the usual way.
- At the device info page an additional line is shown, containing the IP address and the connection state to the remote M&C.

Device window pages

The following table shows which device window pages are available with the Proxy-Device as long as it is unconfigured. As mentioned above, the Proxy-Device will provide the device window pages of the remote device as soon it is connected.

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the Proxy-Device you set the IP address of the remote sat-nms M&C and the name of the device to mirror. The setup page is still accessible when the Proxy-Device is connected, hence you are able to re-configure the destination of a Proxy-Device at any time.

parameter	***description***
address	The proxy device expects the destination address built as '*<IP address>:<device name>* '. For example, '*168.192.1.1:S01* ' refers to a device called 'S01' managed by the M&C at the address 192.168.1.1.

Remarks / Limitations

- The Proxy-Device requires a permanent network link to the remote sat-nms system. Dial-Up connections are not suitable. Also should the network link provide a reasonable bandwidth, 64kbps are the absolute minimum.
- The Proxy-Device is not capable to mirror devices creating variables which are not listed in the device driver file at runtime. Such variables will be silently ignored by the Proxy-Device.
- For the Proxy-Device it is very important that the device drivers both at the local and the remote M&C define exactly the same variables. If some variables appear to not updates as expected, check the device driver versions on both machines.
- When the local M&C boots, the Proxy-Device does not know which type of device it shall mirror until it is connected to the remote system for a couple of seconds. During this time, the proxy device does not provide any variables of the remote device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.proxy.state	TEXT	R/O	StringRange R/O
address	TEXT	SAVE	StringRange
faults.98	ALARM	R/O	AlarmFlagRange R/O (Remote connection)

9.542 Qbit-Q572-AudioReceiver

This page describes the *Qbit-Q572-AudioReceiver* device driver and its device window. The *Qbit-Q572-AudioReceiver* device driver monitors and controls the audio satellite receiver.

The driver automatically recognizes the number of licensed audio decoders within tghe unit. It is prepared to support up to 4 decoders within a single unit.

The driver was tested with the following models:

- Model: Q572 V2.247d (D4 DVBS2 L2 AAC LPCM TS)
- Build: 20190606-114014 SVN:35845_20180920-095639
- Control: 17.12.2015 20:03 1.463.051 ctrl_v01_16.xsvf
- Stream: 14.11.2016 11:35 1.462.932 stream_v01_19.xsv
- Codec: 14.11.2016 11:35 1.462.932 codec_v01_14.xsvf

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The RF monitoring and control parameters.
- --- The audio decoder output monitoring and control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read

and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- audio.1-4.limitLow --- Sets the silence detection level alarm threshold for decoder 1-4
- audio.1-4.limitHigh --- Sets the overload level alarm threshold for decoder 1-4
- audio.1-4.lowT1 --- Sets the silence detection level limit alarm raise delay.
- audio.1-4.highT1 --- Sets the overload level alarm raise delay.
- audio.1-4.lowT2 --- Sets the silence detection level clear delay.
- audio.1-4.highT2 --- Sets the overload detection level clear delay.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter below becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- lnbVoltage --- Selects if the IRD shall apply LNB power to the input as well as the voltage level.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.audio.1.limitLow	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.1.limitHigh	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.1.lowT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.1.highT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.1.lowT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.1.highT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.2.limitLow	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.2.limitHigh	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.2.lowT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.2.highT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.2.lowT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.2.highT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.3.limitLow	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.3.limitHigh	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.audio.3.lowT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.3.highT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.3.lowT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.3.highT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.4.limitLow	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.4.limitHigh	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.audio.4.lowT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.4.highT1	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.4.lowT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.audio.4.highT2	INTEGER	SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SETUP	IntegerRange (0 .. 0)
config.loFreqHigh	INTEGER	SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 12V 13V 18V)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.ber	TEXT	R/O	StringRange R/O
internal.band	TEXT	R/O	StringRange R/O
internal.save	BOOLEAN	R/O	BooleanRange R/O (true false)
interface	CHOICE		EnumRange (ASI RTP UDP ISDN X.21 ES-RTP SAT UDP ES-UDRTP ES-CARD UNKNOWN)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
polarization	CHOICE		EnumRange (X Y)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
dvbmode	CHOICE	R/O	EnumRange R/O (DVB-S DVB-S2 DSS TURBO)
fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 UNKNOWN)
modulation	CHOICE	R/O	EnumRange R/O (QPSK 8PSK 16-APSK 32-APSK UNKNOWN)
pilots	CHOICE	R/O	EnumRange R/O (ON OFF)
frameLength	CHOICE	R/O	EnumRange R/O (LONG SHORT)
rolloff	CHOICE	R/O	EnumRange R/O (35% 25% 20%)
spectrumSense	CHOICE	R/O	EnumRange R/O (NORMAL INVERTED)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ber	TEXT	R/O	StringRange R/O
audio.1.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.active	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.state	CHOICE	R/O	EnumRange R/O (WAITING RECEIVE LOCKED ERROR UNKNOWN)
audio.1.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.1.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.1.sampling	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.1.channel	CHOICE	R/O	EnumRange R/O (NO L R L+R)
audio.2.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.active	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.state	CHOICE	R/O	EnumRange R/O (WAITING RECEIVE LOCKED ERROR UNKNOWN)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.2.sampling	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.2.channel	CHOICE	R/O	EnumRange R/O (NO L R L+R)
audio.3.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.active	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.state	CHOICE	R/O	EnumRange R/O (WAITING RECEIVE LOCKED ERROR UNKNOWN)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.3.sampling	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.3.channel	CHOICE	R/O	EnumRange R/O (NO L R L+R)
audio.4.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.active	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.state	CHOICE	R/O	EnumRange R/O (WAITING RECEIVE LOCKED ERROR UNKNOWN)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
audio.4.sampling	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
audio.4.channel	CHOICE	R/O	EnumRange R/O (NO L R L+R)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.delay	INTEGER		IntegerRange (0 .. 0)
audio.1.output	CHOICE		EnumRange (ANALOG AES-EBU)
audio.1.mute	CHOICE		EnumRange (NONE MUTE AESOFF AESOFF-IP)
audio.1.offset	CHOICE		EnumRange (-12dB -11.5dB -11dB -10.5dB -10dB -9.5dB -9dB -8.5dB -8dB -7.5dB -7dB -6.5dB -6dB -5.5dB -5dB -4.5dB -4dB -3.5dB -3dB -2.5dB -2dB -1.5dB -1dB -0.5dB 0dB +0.5dB +1dB +1.5dB +2dB +2.5dB +3dB +3.5dB +4dB +4.5dB +5dB +5.5dB +6dB +6.5dB +7dB +7.5dB +8dB +8.5dB +9dB +9.5dB +10dB +10.5dB +11dB +11.5dB +12dB)
audio.1.delay2	INTEGER		IntegerRange (0 .. 0)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.delay	INTEGER		IntegerRange (0 .. 0)
audio.2.output	CHOICE		EnumRange (ANALOG AES-EBU)
audio.2.mute	CHOICE		EnumRange (NONE MUTE AESOFF AESOFF-IP)
audio.2.offset	CHOICE		EnumRange (-12dB -11.5dB -11dB -10.5dB -10dB -9.5dB -9dB -8.5dB -8dB -7.5dB -7dB -6.5dB -6dB -5.5dB -5dB -4.5dB -4dB -3.5dB -3dB -2.5dB -2dB -1.5dB -1dB -0.5dB 0dB +0.5dB +1dB +1.5dB +2dB +2.5dB +3dB +3.5dB +4dB +4.5dB +5dB +5.5dB +6dB +6.5dB +7dB +7.5dB +8dB +8.5dB +9dB +9.5dB +10dB +10.5dB +11dB +11.5dB +12dB)
audio.2.delay2	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.delay	INTEGER		IntegerRange (0 .. 0)
audio.3.output	CHOICE		EnumRange (ANALOG AES-EBU)
audio.3.mute	CHOICE		EnumRange (NONE MUTE AESOFF AESOFF-IP)
audio.3.offset	CHOICE		EnumRange (-12dB -11.5dB -11dB -10.5dB -10dB -9.5dB -9dB -8.5dB -8dB -7.5dB -7dB -6.5dB -6dB -5.5dB -5dB -4.5dB -4dB -3.5dB -3dB -2.5dB -2dB -1.5dB -1dB -0.5dB 0dB +0.5dB +1dB +1.5dB +2dB +2.5dB +3dB +3.5dB +4dB +4.5dB +5dB +5.5dB +6dB +6.5dB +7dB +7.5dB +8dB +8.5dB +9dB +9.5dB +10dB +10.5dB +11dB +11.5dB +12dB)
audio.3.delay2	INTEGER		IntegerRange (0 .. 0)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.delay	INTEGER		IntegerRange (0 .. 0)
audio.4.output	CHOICE		EnumRange (ANALOG AES-EBU)
audio.4.mute	CHOICE		EnumRange (NONE MUTE AESOFF AESOFF-IP)
audio.4.offset	CHOICE		EnumRange (-12dB -11.5dB -11dB -10.5dB -10dB -9.5dB -9dB -8.5dB -8dB -7.5dB -7dB -6.5dB -6dB -5.5dB -5dB -4.5dB -4dB -3.5dB -3dB -2.5dB -2dB -1.5dB -1dB -0.5dB 0dB +0.5dB +1dB +1.5dB +2dB +2.5dB +3dB +3.5dB +4dB +4.5dB +5dB +5.5dB +6dB +6.5dB +7dB +7.5dB +8dB +8.5dB +9dB +9.5dB +10dB +10.5dB +11dB +11.5dB +12dB)
audio.4.delay2	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio 1)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 4)

9.543 RFDesign-Amp

This page describes the device driver and the device window for the RFDesign HQS Splitter with redundant amplifier - Single Module

The driver was tested with the following release:

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
info.moduleType	TEXT	R/O	StringRange R/O
info.moduleName	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 13V 15V 18V 13V+22kHz 18V+22kHz)
config.lnbRefLevel	INTEGER	SETUP	IntegerRange (50 .. 350)
config.lnbHysteresis	INTEGER	SETUP	IntegerRange (10 .. 50)
config.agc	CHOICE	SETUP	EnumRange (ON OFF)
config.agcRefLevel	INTEGER	SETUP	IntegerRange (-40 .. 10)
config.agcHysteresis	INTEGER	SETUP	IntegerRange (1 .. 9)
config.agcTimeWindow	INTEGER	SETUP	IntegerRange (1 .. 9)
config.threshold	CHOICE	SETUP	EnumRange (ON OFF)
config.thresholdLevel	INTEGER	SETUP	IntegerRange (-40 .. 10)
config.limiter	CHOICE	SETUP	EnumRange (ON OFF)
gain	INTEGER		IntegerRange (0 .. 25)
eq	INTEGER		IntegerRange (0 .. 9)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
limiterLevel	INTEGER	R/O	IntegerRange R/O (-30 .. 10)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module status)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RF power overflow)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Limiter)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Threshold)
faults.08	ALARM	R/O	AlarmFlagRange R/O (LNB power)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (AGC Lower limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AGC Upper limit)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AGC Lock)

9.544 RFDesign-FLCR-ODA

This page describes the device driver and the device window for the RFDesign optical link chassis optical link chassis FLCR-ODA via SNMP. It uses RFDesign MIB 12.

It was tested with 1:1 model FLCR-41611

- FW: 2.3.33.D20.S31.H23.26.20.4

The driver is capable to monitor/control up to 8 pairs of optical receiver/transmitter modules.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
noSlots	Defines if unit is equipped with 8 or 16 modules/slots
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.noSlots	CHOICE	SAVE SETUP	EnumRange (8 16)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.modStat	TEXT	R/O	StringRange R/O
slot.01.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.01.note	TEXT		StringRange DIS
slot.01.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.01.agc	CHOICE		EnumRange DIS (OFF ON)
slot.01.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.01.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.02.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.02.note	TEXT		StringRange DIS
slot.02.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.02.agc	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
slot.02.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.02.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.03.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.03.note	TEXT		StringRange DIS
slot.03.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.03.agc	CHOICE		EnumRange DIS (OFF ON)
slot.03.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.03.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.03.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.04.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.04.note	TEXT		StringRange DIS
slot.04.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.04.agc	CHOICE		EnumRange DIS (OFF ON)
slot.04.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.04.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.activeAmp	CHOICE		EnumRange DIS (0 1 2)

name	*type*	*flags*	*range*
slot.05.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.05.note	TEXT		StringRange DIS
slot.05.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.05.agc	CHOICE		EnumRange DIS (OFF ON)
slot.05.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.05.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.06.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.06.note	TEXT		StringRange DIS
slot.06.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.06.agc	CHOICE		EnumRange DIS (OFF ON)
slot.06.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.06.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.06.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.07.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.07.note	TEXT		StringRange DIS
slot.07.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.07.agc	CHOICE		EnumRange DIS (OFF ON)
slot.07.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.07.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.07.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.08.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.08.note	TEXT		StringRange DIS
slot.08.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.08.agc	CHOICE		EnumRange DIS (OFF ON)
slot.08.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.08.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.09.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.09.note	TEXT		StringRange DIS
slot.09.gain	INTEGER		IntegerRange DIS (0 .. 31)

name	*type*	*flags*	*range*
slot.09.agc	CHOICE		EnumRange DIS (OFF ON)
slot.09.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.09.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.10.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.10.note	TEXT		StringRange DIS
slot.10.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.10.agc	CHOICE		EnumRange DIS (OFF ON)
slot.10.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.10.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.10.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.11.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.11.note	TEXT		StringRange DIS
slot.11.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.11.agc	CHOICE		EnumRange DIS (OFF ON)
slot.11.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.11.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.activeAmp	CHOICE		EnumRange DIS (0 1 2)

name	*type*	*flags*	*range*
slot.12.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.12.note	TEXT		StringRange DIS
slot.12.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.12.agc	CHOICE		EnumRange DIS (OFF ON)
slot.12.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.12.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.13.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.13.note	TEXT		StringRange DIS
slot.13.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.13.agc	CHOICE		EnumRange DIS (OFF ON)
slot.13.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)

name	*type*	*flags*	*range*
slot.13.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.14.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.14.note	TEXT		StringRange DIS
slot.14.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.14.agc	CHOICE		EnumRange DIS (OFF ON)
slot.14.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.14.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.14.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.15.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.15.note	TEXT		StringRange DIS
slot.15.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.15.agc	CHOICE		EnumRange DIS (OFF ON)
slot.15.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.15.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.16.card	CHOICE	R/O	EnumRange R/O (N/A SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.16.note	TEXT		StringRange DIS

name	*type*	*flags*	*range*
slot.16.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.16.agc	CHOICE		EnumRange DIS (OFF ON)
slot.16.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.16.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.activeAmp	CHOICE		EnumRange DIS (0 1 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 01)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 02)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 03)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 04)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 05)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 06)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 07)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 08)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 09)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 11)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 12)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 13)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 14)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 15)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 16)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.545 RFDesign-FLCR

This page describes the device driver and the device window for the RFDesign optical link chassis RedLink.

The driver is capable to monitor/control up to 16 redundant pairs of optical transmitter modules

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.chassisType	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
internal.modStat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot.01.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.01.note	TEXT		StringRange DIS
slot.01.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.01.agc	CHOICE		EnumRange DIS (OFF ON)
slot.01.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.01.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.02.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.02.note	TEXT		StringRange DIS
slot.02.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.02.agc	CHOICE		EnumRange DIS (OFF ON)
slot.02.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)

name	*type*	*flags*	*range*
slot.02.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.03.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.03.note	TEXT		StringRange DIS
slot.03.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.03.agc	CHOICE		EnumRange DIS (OFF ON)
slot.03.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.03.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.03.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.04.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.04.note	TEXT		StringRange DIS
slot.04.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.04.agc	CHOICE		EnumRange DIS (OFF ON)
slot.04.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.04.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.05.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)

name	*type*	*flags*	*range*
slot.05.note	TEXT		StringRange DIS
slot.05.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.05.agc	CHOICE		EnumRange DIS (OFF ON)
slot.05.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.05.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.06.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.06.note	TEXT		StringRange DIS
slot.06.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.06.agc	CHOICE		EnumRange DIS (OFF ON)
slot.06.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.06.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.06.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.07.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.07.note	TEXT		StringRange DIS
slot.07.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.07.agc	CHOICE		EnumRange DIS (OFF ON)
slot.07.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.07.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.07.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.08.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.08.note	TEXT		StringRange DIS
slot.08.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.08.agc	CHOICE		EnumRange DIS (OFF ON)
slot.08.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.08.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.09.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.09.note	TEXT		StringRange DIS
slot.09.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.09.agc	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
slot.09.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.09.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.10.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.10.note	TEXT		StringRange DIS
slot.10.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.10.agc	CHOICE		EnumRange DIS (OFF ON)
slot.10.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.10.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.10.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.11.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.11.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.11.note	TEXT		StringRange DIS
slot.11.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.11.agc	CHOICE		EnumRange DIS (OFF ON)
slot.11.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.11.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.activeAmp	CHOICE		EnumRange DIS (0 1 2)

name	*type*	*flags*	*range*
slot.12.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.12.note	TEXT		StringRange DIS
slot.12.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.12.agc	CHOICE		EnumRange DIS (OFF ON)
slot.12.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.12.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.13.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.13.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.13.note	TEXT		StringRange DIS
slot.13.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.13.agc	CHOICE		EnumRange DIS (OFF ON)
slot.13.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)

name	*type*	*flags*	*range*
slot.13.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.14.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.14.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.14.note	TEXT		StringRange DIS
slot.14.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.14.agc	CHOICE		EnumRange DIS (OFF ON)
slot.14.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.14.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.14.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.15.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.15.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.15.note	TEXT		StringRange DIS
slot.15.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.15.agc	CHOICE		EnumRange DIS (OFF ON)
slot.15.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.15.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.16.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.16.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)

name	*type*	*flags*	*range*
slot.16.note	TEXT		StringRange DIS
slot.16.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.16.agc	CHOICE		EnumRange DIS (OFF ON)
slot.16.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.16.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.17.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.17.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.17.note	TEXT		StringRange DIS
slot.17.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.17.agc	CHOICE		EnumRange DIS (OFF ON)
slot.17.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.17.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.17.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.17.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.17.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.18.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.18.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.18.note	TEXT		StringRange DIS
slot.18.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.18.agc	CHOICE		EnumRange DIS (OFF ON)
slot.18.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.18.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.18.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.19.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.19.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.19.note	TEXT		StringRange DIS
slot.19.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.19.agc	CHOICE		EnumRange DIS (OFF ON)
slot.19.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.19.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.19.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.19.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.19.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.20.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.20.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.20.note	TEXT		StringRange DIS
slot.20.gain	INTEGER		IntegerRange DIS (0 .. 31)

name	*type*	*flags*	*range*
slot.20.agc	CHOICE		EnumRange DIS (OFF ON)
slot.20.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.20.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.21.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.21.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.21.note	TEXT		StringRange DIS
slot.21.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.21.agc	CHOICE		EnumRange DIS (OFF ON)
slot.21.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.21.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.21.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.22.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.22.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.22.note	TEXT		StringRange DIS
slot.22.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.22.agc	CHOICE		EnumRange DIS (OFF ON)
slot.22.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.22.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.22.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.22.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.22.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.activeAmp	CHOICE		EnumRange DIS (0 1 2)

name	*type*	*flags*	*range*
slot.23.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.23.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.23.note	TEXT		StringRange DIS
slot.23.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.23.agc	CHOICE		EnumRange DIS (OFF ON)
slot.23.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.23.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.24.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.24.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.24.note	TEXT		StringRange DIS
slot.24.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.24.agc	CHOICE		EnumRange DIS (OFF ON)
slot.24.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)

name	*type*	*flags*	*range*
slot.24.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.25.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.25.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.25.note	TEXT		StringRange DIS
slot.25.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.25.agc	CHOICE		EnumRange DIS (OFF ON)
slot.25.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.25.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.25.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.25.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.25.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.25.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.26.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.26.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.26.note	TEXT		StringRange DIS
slot.26.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.26.agc	CHOICE		EnumRange DIS (OFF ON)
slot.26.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.26.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.27.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.27.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)

name	*type*	*flags*	*range*
slot.27.note	TEXT		StringRange DIS
slot.27.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.27.agc	CHOICE		EnumRange DIS (OFF ON)
slot.27.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.27.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.28.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.28.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.28.note	TEXT		StringRange DIS
slot.28.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.28.agc	CHOICE		EnumRange DIS (OFF ON)
slot.28.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.28.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.28.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.28.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.28.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.29.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.29.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.29.note	TEXT		StringRange DIS
slot.29.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.29.agc	CHOICE		EnumRange DIS (OFF ON)
slot.29.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.29.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.29.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.30.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.30.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.30.note	TEXT		StringRange DIS
slot.30.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.30.agc	CHOICE		EnumRange DIS (OFF ON)
slot.30.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.30.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.31.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.31.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.31.note	TEXT		StringRange DIS
slot.31.gain	INTEGER		IntegerRange DIS (0 .. 31)

name	*type*	*flags*	*range*
slot.31.agc	CHOICE		EnumRange DIS (OFF ON)
slot.31.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.31.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.32.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.32.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.32.note	TEXT		StringRange DIS
slot.32.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.32.agc	CHOICE		EnumRange DIS (OFF ON)
slot.32.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.32.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.32.rxFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.activeAmp	CHOICE		EnumRange DIS (0 1 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 01)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 02)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 03)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 04)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 05)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 06)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 07)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 08)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 09)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 11)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 12)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 13)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 14)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 15)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 16)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 17)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 18)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 19)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 20)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 21)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 22)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 23)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 24)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 25)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 26)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 27)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 28)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 29)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 30)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 31)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 32)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.546 RFDesign-FlexLink-S7B

This page describes the device driver and the device window for RF Design L-Band Switch Matrix S7B.

It supports 8x16 and 16x16 in and outputs.
It was tested with SW: 3.1.A13.0.D3.2.S6.H8.8

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-16.
- 1 2 3 --- Port switching by number 1-16.
- --- Input state, alias names and configuration.

- --- Edit output alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- *readBackDelay* --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.choice	TEXT	R/O	StringRange R/O
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.posName	TEXT	R/O	StringRange R/O
aliasRead	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
output.01	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.02	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.03	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.04	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
output.05	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.06	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.07	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.08	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.09	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.10	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.11	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.12	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.13	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.14	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.15	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
output.16	CHOICE		EnumRange (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.01	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.02	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.03	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.04	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)

name	*type*	*flags*	*range*
byname.05	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.06	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.07	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.08	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.09	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.10	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.11	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.12	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.13	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.14	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.15	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
byname.16	CHOICE	NOPRESET	EnumRange (0 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16)
rf.level.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rf.level.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.current.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gain.01	INTEGER		IntegerRange (0 .. 10)

name	*type*	*flags*	*range*
gain.02	INTEGER		IntegerRange (0 .. 10)
gain.03	INTEGER		IntegerRange (0 .. 10)
gain.04	INTEGER		IntegerRange (0 .. 10)
gain.05	INTEGER		IntegerRange (0 .. 10)
gain.06	INTEGER		IntegerRange (0 .. 10)
gain.07	INTEGER		IntegerRange (0 .. 10)
gain.08	INTEGER		IntegerRange (0 .. 10)
gain.09	INTEGER		IntegerRange (0 .. 10)
gain.10	INTEGER		IntegerRange (0 .. 10)
gain.11	INTEGER		IntegerRange (0 .. 10)
gain.12	INTEGER		IntegerRange (0 .. 10)
gain.13	INTEGER		IntegerRange (0 .. 10)
gain.14	INTEGER		IntegerRange (0 .. 10)
gain.15	INTEGER		IntegerRange (0 .. 10)
gain.16	INTEGER		IntegerRange (0 .. 10)
slope.01	INTEGER		IntegerRange (0 .. 9)
slope.02	INTEGER		IntegerRange (0 .. 9)
slope.03	INTEGER		IntegerRange (0 .. 9)
slope.04	INTEGER		IntegerRange (0 .. 9)
slope.05	INTEGER		IntegerRange (0 .. 9)
slope.06	INTEGER		IntegerRange (0 .. 9)
slope.07	INTEGER		IntegerRange (0 .. 9)
slope.08	INTEGER		IntegerRange (0 .. 9)
slope.09	INTEGER		IntegerRange (0 .. 9)
slope.10	INTEGER		IntegerRange (0 .. 9)
slope.11	INTEGER		IntegerRange (0 .. 9)

name	*type*	*flags*	*range*
slope.12	INTEGER		IntegerRange (0 .. 9)
slope.13	INTEGER		IntegerRange (0 .. 9)
slope.14	INTEGER		IntegerRange (0 .. 9)
slope.15	INTEGER		IntegerRange (0 .. 9)
slope.16	INTEGER		IntegerRange (0 .. 9)
thd.01	INTEGER		IntegerRange (-40 .. 10)
thd.02	INTEGER		IntegerRange (-40 .. 10)
thd.03	INTEGER		IntegerRange (-40 .. 10)
thd.04	INTEGER		IntegerRange (-40 .. 10)
thd.05	INTEGER		IntegerRange (-40 .. 10)
thd.06	INTEGER		IntegerRange (-40 .. 10)
thd.07	INTEGER		IntegerRange (-40 .. 10)
thd.08	INTEGER		IntegerRange (-40 .. 10)
thd.09	INTEGER		IntegerRange (-40 .. 10)
thd.10	INTEGER		IntegerRange (-40 .. 10)
thd.11	INTEGER		IntegerRange (-40 .. 10)
thd.12	INTEGER		IntegerRange (-40 .. 10)
thd.13	INTEGER		IntegerRange (-40 .. 10)
thd.14	INTEGER		IntegerRange (-40 .. 10)
thd.15	INTEGER		IntegerRange (-40 .. 10)
thd.16	INTEGER		IntegerRange (-40 .. 10)
thd.enable.01	CHOICE		EnumRange (ON OFF)
thd.enable.02	CHOICE		EnumRange (ON OFF)
thd.enable.03	CHOICE		EnumRange (ON OFF)
thd.enable.04	CHOICE		EnumRange (ON OFF)
thd.enable.05	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
thd.enable.06	CHOICE		EnumRange (ON OFF)
thd.enable.07	CHOICE		EnumRange (ON OFF)
thd.enable.08	CHOICE		EnumRange (ON OFF)
thd.enable.09	CHOICE		EnumRange (ON OFF)
thd.enable.10	CHOICE		EnumRange (ON OFF)
thd.enable.11	CHOICE		EnumRange (ON OFF)
thd.enable.12	CHOICE		EnumRange (ON OFF)
thd.enable.13	CHOICE		EnumRange (ON OFF)
thd.enable.14	CHOICE		EnumRange (ON OFF)
thd.enable.15	CHOICE		EnumRange (ON OFF)
thd.enable.16	CHOICE		EnumRange (ON OFF)
thd.hysteresis.01	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.02	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.03	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.04	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.05	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.06	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.07	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.08	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.09	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.10	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.11	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.12	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.13	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.14	INTEGER		IntegerRange (0 .. 9)
thd.hysteresis.15	INTEGER		IntegerRange (0 .. 9)

name	*type*	*flags*	*range*
thd.hysteresis.16	INTEGER		IntegerRange (0 .. 9)
lnb.power.01	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.02	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.03	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.04	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.05	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.06	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.07	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.08	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.09	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.10	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.11	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.12	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.13	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.14	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.15	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)
lnb.power.16	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22KHz 18V+22KHz)

name	*type*	*flags*	*range*
lnb.threshold.01	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.02	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.03	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.04	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.05	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.06	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.07	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.08	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.09	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.10	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.11	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.12	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.13	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.14	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.15	INTEGER		IntegerRange (50 .. 400)
lnb.threshold.16	INTEGER		IntegerRange (50 .. 400)
lnb.hysteresis.01	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.02	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.03	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.04	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.05	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.06	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.07	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.08	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.09	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.10	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.11	INTEGER		IntegerRange (10 .. 50)

name	*type*	*flags*	*range*
lnb.hysteresis.12	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.13	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.14	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.15	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.16	INTEGER		IntegerRange (10 .. 50)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit Warning)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Input 8)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input 9)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input 15)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (Input 16)
faults.31	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 2)
faults.33	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 3)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 4)
faults.35	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 5)
faults.36	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 6)
faults.37	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 7)
faults.38	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 8)
faults.39	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 9)
faults.40	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 10)
faults.41	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 11)
faults.42	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 12)
faults.43	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 13)
faults.44	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 14)
faults.45	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 15)
faults.46	ALARM	R/O	AlarmFlagRange R/O (RF Level Input 16)
faults.51	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 1)
faults.52	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 2)
faults.53	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 3)

name	*type*	*flags*	*range*
faults.54	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 4)
faults.55	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 5)
faults.56	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 6)
faults.57	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 7)
faults.58	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 8)
faults.59	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 9)
faults.60	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 10)
faults.61	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 11)
faults.62	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 12)
faults.63	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 13)
faults.64	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 14)
faults.65	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 15)
faults.66	ALARM	R/O	AlarmFlagRange R/O (LNB Supply Input 16)

9.547 RFDesign-FlexLink-S7Bc

This page describes the device driver and the device window for RF Design Combining L-Band Matrix S7Bc.

It supports up to 16 in and 8 outputs.
It was tested with SW: 3.1.A2.0.D1.1.S1.H1.1

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Select Crosspoints.
- --- Output state, alias names and configuration.
- --- Edit input alias names.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- *readBackDelay* --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Alias names

After changing the alias names of the input and or output channels from the device driver or Web interface it is necessary to use the read alias names from chassis button on the device driver. The reason is that reading these value is only done initially at server startup due to the slow device SNMP performance.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.pos	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aliasRead	TEXT	NOPRESET	StringRange
output.01.name	TEXT		StringRange
output.02.name	TEXT		StringRange
output.03.name	TEXT		StringRange
output.04.name	TEXT		StringRange
output.05.name	TEXT		StringRange
output.06.name	TEXT		StringRange
output.07.name	TEXT		StringRange
output.08.name	TEXT		StringRange
output.09.name	TEXT		StringRange
output.10.name	TEXT		StringRange

name	*type*	*flags*	*range*
output.11.name	TEXT		StringRange
output.12.name	TEXT		StringRange
output.13.name	TEXT		StringRange
output.14.name	TEXT		StringRange
output.15.name	TEXT		StringRange
output.16.name	TEXT		StringRange
input.01.name	TEXT		StringRange
input.02.name	TEXT		StringRange
input.03.name	TEXT		StringRange
input.04.name	TEXT		StringRange
input.05.name	TEXT		StringRange
input.06.name	TEXT		StringRange
input.07.name	TEXT		StringRange
input.08.name	TEXT		StringRange
input.09.name	TEXT		StringRange
input.10.name	TEXT		StringRange
input.11.name	TEXT		StringRange
input.12.name	TEXT		StringRange
input.13.name	TEXT		StringRange
input.14.name	TEXT		StringRange
input.15.name	TEXT		StringRange
input.16.name	TEXT		StringRange
out.01.in.01	CHOICE		EnumRange (OFF ON)
out.01.in.02	CHOICE		EnumRange (OFF ON)
out.01.in.03	CHOICE		EnumRange (OFF ON)
out.01.in.04	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.01.in.05	CHOICE		EnumRange (OFF ON)
out.01.in.06	CHOICE		EnumRange (OFF ON)
out.01.in.07	CHOICE		EnumRange (OFF ON)
out.01.in.08	CHOICE		EnumRange (OFF ON)
out.01.in.09	CHOICE		EnumRange (OFF ON)
out.01.in.10	CHOICE		EnumRange (OFF ON)
out.01.in.11	CHOICE		EnumRange (OFF ON)
out.01.in.12	CHOICE		EnumRange (OFF ON)
out.01.in.13	CHOICE		EnumRange (OFF ON)
out.01.in.14	CHOICE		EnumRange (OFF ON)
out.01.in.15	CHOICE		EnumRange (OFF ON)
out.01.in.16	CHOICE		EnumRange (OFF ON)
out.02.in.01	CHOICE		EnumRange (OFF ON)
out.02.in.02	CHOICE		EnumRange (OFF ON)
out.02.in.03	CHOICE		EnumRange (OFF ON)
out.02.in.04	CHOICE		EnumRange (OFF ON)
out.02.in.05	CHOICE		EnumRange (OFF ON)
out.02.in.06	CHOICE		EnumRange (OFF ON)
out.02.in.07	CHOICE		EnumRange (OFF ON)
out.02.in.08	CHOICE		EnumRange (OFF ON)
out.02.in.09	CHOICE		EnumRange (OFF ON)
out.02.in.10	CHOICE		EnumRange (OFF ON)
out.02.in.11	CHOICE		EnumRange (OFF ON)
out.02.in.12	CHOICE		EnumRange (OFF ON)
out.02.in.13	CHOICE		EnumRange (OFF ON)
out.02.in.14	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.02.in.15	CHOICE		EnumRange (OFF ON)
out.02.in.16	CHOICE		EnumRange (OFF ON)
out.03.in.01	CHOICE		EnumRange (OFF ON)
out.03.in.02	CHOICE		EnumRange (OFF ON)
out.03.in.03	CHOICE		EnumRange (OFF ON)
out.03.in.04	CHOICE		EnumRange (OFF ON)
out.03.in.05	CHOICE		EnumRange (OFF ON)
out.03.in.06	CHOICE		EnumRange (OFF ON)
out.03.in.07	CHOICE		EnumRange (OFF ON)
out.03.in.08	CHOICE		EnumRange (OFF ON)
out.03.in.09	CHOICE		EnumRange (OFF ON)
out.03.in.10	CHOICE		EnumRange (OFF ON)
out.03.in.11	CHOICE		EnumRange (OFF ON)
out.03.in.12	CHOICE		EnumRange (OFF ON)
out.03.in.13	CHOICE		EnumRange (OFF ON)
out.03.in.14	CHOICE		EnumRange (OFF ON)
out.03.in.15	CHOICE		EnumRange (OFF ON)
out.03.in.16	CHOICE		EnumRange (OFF ON)
out.04.in.01	CHOICE		EnumRange (OFF ON)
out.04.in.02	CHOICE		EnumRange (OFF ON)
out.04.in.03	CHOICE		EnumRange (OFF ON)
out.04.in.04	CHOICE		EnumRange (OFF ON)
out.04.in.05	CHOICE		EnumRange (OFF ON)
out.04.in.06	CHOICE		EnumRange (OFF ON)
out.04.in.07	CHOICE		EnumRange (OFF ON)
out.04.in.08	CHOICE		EnumRange (OFF ON)
out.04.in.09	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.04.in.10	CHOICE		EnumRange (OFF ON)
out.04.in.11	CHOICE		EnumRange (OFF ON)
out.04.in.12	CHOICE		EnumRange (OFF ON)
out.04.in.13	CHOICE		EnumRange (OFF ON)
out.04.in.14	CHOICE		EnumRange (OFF ON)
out.04.in.15	CHOICE		EnumRange (OFF ON)
out.04.in.16	CHOICE		EnumRange (OFF ON)
out.05.in.01	CHOICE		EnumRange (OFF ON)
out.05.in.02	CHOICE		EnumRange (OFF ON)
out.05.in.03	CHOICE		EnumRange (OFF ON)
out.05.in.04	CHOICE		EnumRange (OFF ON)
out.05.in.05	CHOICE		EnumRange (OFF ON)
out.05.in.06	CHOICE		EnumRange (OFF ON)
out.05.in.07	CHOICE		EnumRange (OFF ON)
out.05.in.08	CHOICE		EnumRange (OFF ON)
out.05.in.09	CHOICE		EnumRange (OFF ON)
out.05.in.10	CHOICE		EnumRange (OFF ON)
out.05.in.11	CHOICE		EnumRange (OFF ON)
out.05.in.12	CHOICE		EnumRange (OFF ON)
out.05.in.13	CHOICE		EnumRange (OFF ON)
out.05.in.14	CHOICE		EnumRange (OFF ON)
out.05.in.15	CHOICE		EnumRange (OFF ON)
out.05.in.16	CHOICE		EnumRange (OFF ON)
out.06.in.01	CHOICE		EnumRange (OFF ON)
out.06.in.02	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.06.in.03	CHOICE		EnumRange (OFF ON)
out.06.in.04	CHOICE		EnumRange (OFF ON)
out.06.in.05	CHOICE		EnumRange (OFF ON)
out.06.in.06	CHOICE		EnumRange (OFF ON)
out.06.in.07	CHOICE		EnumRange (OFF ON)
out.06.in.08	CHOICE		EnumRange (OFF ON)
out.06.in.09	CHOICE		EnumRange (OFF ON)
out.06.in.10	CHOICE		EnumRange (OFF ON)
out.06.in.11	CHOICE		EnumRange (OFF ON)
out.06.in.12	CHOICE		EnumRange (OFF ON)
out.06.in.13	CHOICE		EnumRange (OFF ON)
out.06.in.14	CHOICE		EnumRange (OFF ON)
out.06.in.15	CHOICE		EnumRange (OFF ON)
out.06.in.16	CHOICE		EnumRange (OFF ON)
out.07.in.01	CHOICE		EnumRange (OFF ON)
out.07.in.02	CHOICE		EnumRange (OFF ON)
out.07.in.03	CHOICE		EnumRange (OFF ON)
out.07.in.04	CHOICE		EnumRange (OFF ON)
out.07.in.05	CHOICE		EnumRange (OFF ON)
out.07.in.06	CHOICE		EnumRange (OFF ON)
out.07.in.07	CHOICE		EnumRange (OFF ON)
out.07.in.08	CHOICE		EnumRange (OFF ON)
out.07.in.09	CHOICE		EnumRange (OFF ON)
out.07.in.10	CHOICE		EnumRange (OFF ON)
out.07.in.11	CHOICE		EnumRange (OFF ON)
out.07.in.12	CHOICE		EnumRange (OFF ON)
out.07.in.13	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
out.07.in.14	CHOICE		EnumRange (OFF ON)
out.07.in.15	CHOICE		EnumRange (OFF ON)
out.07.in.16	CHOICE		EnumRange (OFF ON)
out.08.in.01	CHOICE		EnumRange (OFF ON)
out.08.in.02	CHOICE		EnumRange (OFF ON)
out.08.in.03	CHOICE		EnumRange (OFF ON)
out.08.in.04	CHOICE		EnumRange (OFF ON)
out.08.in.05	CHOICE		EnumRange (OFF ON)
out.08.in.06	CHOICE		EnumRange (OFF ON)
out.08.in.07	CHOICE		EnumRange (OFF ON)
out.08.in.08	CHOICE		EnumRange (OFF ON)
out.08.in.09	CHOICE		EnumRange (OFF ON)
out.08.in.10	CHOICE		EnumRange (OFF ON)
out.08.in.11	CHOICE		EnumRange (OFF ON)
out.08.in.12	CHOICE		EnumRange (OFF ON)
out.08.in.13	CHOICE		EnumRange (OFF ON)
out.08.in.14	CHOICE		EnumRange (OFF ON)
out.08.in.15	CHOICE		EnumRange (OFF ON)
out.08.in.16	CHOICE		EnumRange (OFF ON)
rf.level.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rf.level.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rf.level.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gain.01	INTEGER		IntegerRange (0 .. 10)
gain.02	INTEGER		IntegerRange (0 .. 10)
gain.03	INTEGER		IntegerRange (0 .. 10)
gain.04	INTEGER		IntegerRange (0 .. 10)
gain.05	INTEGER		IntegerRange (0 .. 10)
gain.06	INTEGER		IntegerRange (0 .. 10)
gain.07	INTEGER		IntegerRange (0 .. 10)
gain.08	INTEGER		IntegerRange (0 .. 10)
thd.01	INTEGER		IntegerRange (-40 .. 10)
thd.02	INTEGER		IntegerRange (-40 .. 10)
thd.03	INTEGER		IntegerRange (-40 .. 10)
thd.04	INTEGER		IntegerRange (-40 .. 10)
thd.05	INTEGER		IntegerRange (-40 .. 10)
thd.06	INTEGER		IntegerRange (-40 .. 10)
thd.07	INTEGER		IntegerRange (-40 .. 10)
thd.08	INTEGER		IntegerRange (-40 .. 10)
thd.enable.01	CHOICE		EnumRange (ON OFF)
thd.enable.02	CHOICE		EnumRange (ON OFF)
thd.enable.03	CHOICE		EnumRange (ON OFF)
thd.enable.04	CHOICE		EnumRange (ON OFF)
thd.enable.05	CHOICE		EnumRange (ON OFF)
thd.enable.06	CHOICE		EnumRange (ON OFF)
thd.enable.07	CHOICE		EnumRange (ON OFF)
thd.enable.08	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit Warning)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.21	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 6)
faults.27	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 7)
faults.28	ALARM	R/O	AlarmFlagRange R/O (RF Level Output 8)

9.548 RFDesign-LSC

This page describes the device driver and the device window for the RFDesign LSC Chassis for Splitter and Combiner Cards with redundant amplifier - Up to 8 Modules

The driver was tested with the following release:

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.slot.01.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.02.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.03.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.04.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)

name	*type*	*flags*	*range*
internal.slot.05.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.06.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.07.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.08.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
internal.slot.01.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.02.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.03.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.04.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.05.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.06.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.07.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
internal.slot.08.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.01.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.02.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.03.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.04.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)

name	*type*	*flags*	*range*
slot.05.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.06.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.07.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.08.type	CHOICE	R/O	EnumRange R/O (EMPTY SPLITTER COMBINER)
slot.01.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.02.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.03.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.04.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.05.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.06.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.07.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.08.state	CHOICE	R/O	EnumRange R/O (OK HW FAULT WARNING MISSING)
slot.01.input.01.name	TEXT	SAVE	StringRange
slot.01.input.02.name	TEXT	SAVE	StringRange
slot.01.input.03.name	TEXT	SAVE	StringRange
slot.01.input.04.name	TEXT	SAVE	StringRange
slot.01.input.05.name	TEXT	SAVE	StringRange
slot.01.input.06.name	TEXT	SAVE	StringRange
slot.01.input.07.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
slot.01.input.08.name	TEXT	SAVE	StringRange
slot.02.input.01.name	TEXT	SAVE	StringRange
slot.02.input.02.name	TEXT	SAVE	StringRange
slot.02.input.03.name	TEXT	SAVE	StringRange
slot.02.input.04.name	TEXT	SAVE	StringRange
slot.02.input.05.name	TEXT	SAVE	StringRange
slot.02.input.06.name	TEXT	SAVE	StringRange
slot.02.input.07.name	TEXT	SAVE	StringRange
slot.02.input.08.name	TEXT	SAVE	StringRange
slot.03.input.01.name	TEXT	SAVE	StringRange
slot.03.input.02.name	TEXT	SAVE	StringRange
slot.03.input.03.name	TEXT	SAVE	StringRange
slot.03.input.04.name	TEXT	SAVE	StringRange
slot.03.input.05.name	TEXT	SAVE	StringRange
slot.03.input.06.name	TEXT	SAVE	StringRange
slot.03.input.07.name	TEXT	SAVE	StringRange
slot.03.input.08.name	TEXT	SAVE	StringRange
slot.04.input.01.name	TEXT	SAVE	StringRange
slot.04.input.02.name	TEXT	SAVE	StringRange
slot.04.input.03.name	TEXT	SAVE	StringRange
slot.04.input.04.name	TEXT	SAVE	StringRange
slot.04.input.05.name	TEXT	SAVE	StringRange
slot.04.input.06.name	TEXT	SAVE	StringRange
slot.04.input.07.name	TEXT	SAVE	StringRange
slot.04.input.08.name	TEXT	SAVE	StringRange
slot.05.input.01.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
slot.05.input.02.name	TEXT	SAVE	StringRange
slot.05.input.03.name	TEXT	SAVE	StringRange
slot.05.input.04.name	TEXT	SAVE	StringRange
slot.05.input.05.name	TEXT	SAVE	StringRange
slot.05.input.06.name	TEXT	SAVE	StringRange
slot.05.input.07.name	TEXT	SAVE	StringRange
slot.05.input.08.name	TEXT	SAVE	StringRange
slot.06.input.01.name	TEXT	SAVE	StringRange
slot.06.input.02.name	TEXT	SAVE	StringRange
slot.06.input.03.name	TEXT	SAVE	StringRange
slot.06.input.04.name	TEXT	SAVE	StringRange
slot.06.input.05.name	TEXT	SAVE	StringRange
slot.06.input.06.name	TEXT	SAVE	StringRange
slot.06.input.07.name	TEXT	SAVE	StringRange
slot.06.input.08.name	TEXT	SAVE	StringRange
slot.07.input.01.name	TEXT	SAVE	StringRange
slot.07.input.02.name	TEXT	SAVE	StringRange
slot.07.input.03.name	TEXT	SAVE	StringRange
slot.07.input.04.name	TEXT	SAVE	StringRange
slot.07.input.05.name	TEXT	SAVE	StringRange
slot.07.input.06.name	TEXT	SAVE	StringRange
slot.07.input.07.name	TEXT	SAVE	StringRange
slot.07.input.08.name	TEXT	SAVE	StringRange
slot.08.input.01.name	TEXT	SAVE	StringRange
slot.08.input.02.name	TEXT	SAVE	StringRange
slot.08.input.03.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
slot.08.input.04.name	TEXT	SAVE	StringRange
slot.08.input.05.name	TEXT	SAVE	StringRange
slot.08.input.06.name	TEXT	SAVE	StringRange
slot.08.input.07.name	TEXT	SAVE	StringRange
slot.08.input.08.name	TEXT	SAVE	StringRange
slot.01.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.amp.position	CHOICE		EnumRange (1 2)
slot.01.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.input.01.on	CHOICE		EnumRange (ON OFF)
slot.01.input.02.on	CHOICE		EnumRange (ON OFF)
slot.01.input.03.on	CHOICE		EnumRange (ON OFF)
slot.01.input.04.on	CHOICE		EnumRange (ON OFF)
slot.01.input.05.on	CHOICE		EnumRange (ON OFF)
slot.01.input.06.on	CHOICE		EnumRange (ON OFF)
slot.01.input.07.on	CHOICE		EnumRange (ON OFF)
slot.01.input.08.on	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
slot.01.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.01.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.01.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.01.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.01.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.02.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.02.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.amp.position	CHOICE		EnumRange (1 2)
slot.02.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.input.01.on	CHOICE		EnumRange (ON OFF)
slot.02.input.02.on	CHOICE		EnumRange (ON OFF)
slot.02.input.03.on	CHOICE		EnumRange (ON OFF)
slot.02.input.04.on	CHOICE		EnumRange (ON OFF)
slot.02.input.05.on	CHOICE		EnumRange (ON OFF)
slot.02.input.06.on	CHOICE		EnumRange (ON OFF)
slot.02.input.07.on	CHOICE		EnumRange (ON OFF)
slot.02.input.08.on	CHOICE		EnumRange (ON OFF)
slot.02.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.06.thd	INTEGER		IntegerRange (-99 .. 99)

name	*type*	*flags*	*range*
slot.02.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.02.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.02.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.02.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.02.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.03.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.amp.position	CHOICE		EnumRange (1 2)
slot.03.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.03.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.input.01.on	CHOICE		EnumRange (ON OFF)
slot.03.input.02.on	CHOICE		EnumRange (ON OFF)
slot.03.input.03.on	CHOICE		EnumRange (ON OFF)
slot.03.input.04.on	CHOICE		EnumRange (ON OFF)
slot.03.input.05.on	CHOICE		EnumRange (ON OFF)
slot.03.input.06.on	CHOICE		EnumRange (ON OFF)
slot.03.input.07.on	CHOICE		EnumRange (ON OFF)
slot.03.input.08.on	CHOICE		EnumRange (ON OFF)
slot.03.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.03.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.05.thdOn	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
slot.03.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.03.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.03.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.03.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.04.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.amp.position	CHOICE		EnumRange (1 2)
slot.04.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.04.input.01.on	CHOICE		EnumRange (ON OFF)
slot.04.input.02.on	CHOICE		EnumRange (ON OFF)
slot.04.input.03.on	CHOICE		EnumRange (ON OFF)
slot.04.input.04.on	CHOICE		EnumRange (ON OFF)
slot.04.input.05.on	CHOICE		EnumRange (ON OFF)
slot.04.input.06.on	CHOICE		EnumRange (ON OFF)
slot.04.input.07.on	CHOICE		EnumRange (ON OFF)
slot.04.input.08.on	CHOICE		EnumRange (ON OFF)
slot.04.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.04.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.04.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.02.clrFault	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
slot.04.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.04.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.04.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.05.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.amp.position	CHOICE		EnumRange (1 2)
slot.05.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.input.01.on	CHOICE		EnumRange (ON OFF)
slot.05.input.02.on	CHOICE		EnumRange (ON OFF)
slot.05.input.03.on	CHOICE		EnumRange (ON OFF)
slot.05.input.04.on	CHOICE		EnumRange (ON OFF)
slot.05.input.05.on	CHOICE		EnumRange (ON OFF)
slot.05.input.06.on	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
slot.05.input.07.on	CHOICE		EnumRange (ON OFF)
slot.05.input.08.on	CHOICE		EnumRange (ON OFF)
slot.05.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.05.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.05.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.05.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.05.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)

name	*type*	*flags*	*range*
slot.06.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.amp.position	CHOICE		EnumRange (1 2)
slot.06.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.input.01.on	CHOICE		EnumRange (ON OFF)
slot.06.input.02.on	CHOICE		EnumRange (ON OFF)
slot.06.input.03.on	CHOICE		EnumRange (ON OFF)
slot.06.input.04.on	CHOICE		EnumRange (ON OFF)
slot.06.input.05.on	CHOICE		EnumRange (ON OFF)
slot.06.input.06.on	CHOICE		EnumRange (ON OFF)
slot.06.input.07.on	CHOICE		EnumRange (ON OFF)
slot.06.input.08.on	CHOICE		EnumRange (ON OFF)
slot.06.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.05.thd	INTEGER		IntegerRange (-99 .. 99)

name	*type*	*flags*	*range*
slot.06.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.06.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.06.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.06.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.06.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.07.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.amp.position	CHOICE		EnumRange (1 2)
slot.07.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.07.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.input.01.on	CHOICE		EnumRange (ON OFF)
slot.07.input.02.on	CHOICE		EnumRange (ON OFF)
slot.07.input.03.on	CHOICE		EnumRange (ON OFF)
slot.07.input.04.on	CHOICE		EnumRange (ON OFF)
slot.07.input.05.on	CHOICE		EnumRange (ON OFF)
slot.07.input.06.on	CHOICE		EnumRange (ON OFF)
slot.07.input.07.on	CHOICE		EnumRange (ON OFF)
slot.07.input.08.on	CHOICE		EnumRange (ON OFF)
slot.07.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.07.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.02.thdOn	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
slot.07.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.08.thdOn	CHOICE		EnumRange (ON OFF)
slot.07.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.07.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.07.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
slot.08.amp.1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.amp.2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.amp.position	CHOICE		EnumRange (1 2)
slot.08.amp.gain	FLOAT		DoubleRange (-10.0 .. 10.0)
slot.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.01.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.02.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.03.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.04.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.05.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.06.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot.08.input.07.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.08.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.input.01.on	CHOICE		EnumRange (ON OFF)
slot.08.input.02.on	CHOICE		EnumRange (ON OFF)
slot.08.input.03.on	CHOICE		EnumRange (ON OFF)
slot.08.input.04.on	CHOICE		EnumRange (ON OFF)
slot.08.input.05.on	CHOICE		EnumRange (ON OFF)
slot.08.input.06.on	CHOICE		EnumRange (ON OFF)
slot.08.input.07.on	CHOICE		EnumRange (ON OFF)
slot.08.input.08.on	CHOICE		EnumRange (ON OFF)
slot.08.input.01.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.02.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.03.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.04.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.05.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.06.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.07.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.08.thd	INTEGER		IntegerRange (-99 .. 99)
slot.08.input.01.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.02.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.03.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.04.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.05.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.06.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.07.thdOn	CHOICE		EnumRange (ON OFF)
slot.08.input.08.thdOn	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
slot.08.input.01.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.02.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.03.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.04.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.05.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.06.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.07.clrFault	TEXT	NOPRESET	StringRange
slot.08.input.08.clrFault	TEXT	NOPRESET	StringRange
slot.08.saPort	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 COM)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 1 Amp 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 1 Amp 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 1 Threshold)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 2 Threshold)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 3 Threshold)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 4 Threshold)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 5 Threshold)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 6 Threshold)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 7 Threshold)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 1 In 8 Threshold)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Amp 1)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Amp 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 1 Threshold)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 2 Threshold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 3 Threshold)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 4 Threshold)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 5 Threshold)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 6 Threshold)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 7 Threshold)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 2 In 8 Threshold)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 3 Amp 1)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 3 Amp 2)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 1 Threshold)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 2 Threshold)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 3 Threshold)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 4 Threshold)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 5 Threshold)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 6 Threshold)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 7 Threshold)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Slot 3 In 8 Threshold)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Slot 4)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Slot 4 Amp 1)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Slot 4 Amp 2)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 1 Threshold)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 2 Threshold)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 3 Threshold)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 4 Threshold)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 5 Threshold)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 6 Threshold)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 7 Threshold)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Slot 4 In 8 Threshold)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Slot 5)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Amp 1)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Amp 2)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 1 Threshold)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 2 Threshold)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 3 Threshold)

name	*type*	*flags*	*range*
faults.51	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 4 Threshold)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 5 Threshold)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 6 Threshold)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 7 Threshold)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Slot 5 In 8 Threshold)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Slot 6)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Slot 6 Amp 1)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Slot 6 Amp 2)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 1 Threshold)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 2 Threshold)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 3 Threshold)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 4 Threshold)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 5 Threshold)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 6 Threshold)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 7 Threshold)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Slot 6 In 8 Threshold)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Slot 7)

name	*type*	*flags*	*range*
faults.68	ALARM	R/O	AlarmFlagRange R/O (Slot 7 Amp 1)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Slot 7 Amp 2)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 1 Threshold)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 2 Threshold)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 3 Threshold)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 4 Threshold)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 5 Threshold)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 6 Threshold)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 7 Threshold)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Slot 7 In 8 Threshold)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Slot 8)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Amp 1)
faults.80	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Amp 2)
faults.81	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 1 Threshold)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 2 Threshold)
faults.83	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 3 Threshold)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 4 Threshold)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 5 Threshold)

name	*type*	*flags*	*range*
faults.86	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 6 Threshold)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 7 Threshold)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Slot 8 In 8 Threshold)
faults.89	ALARM	R/O	AlarmFlagRange R/O (Unit status)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.91	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.92	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.549 RFDesign-QLink

This page describes the device driver and the device window for the RFDesign optical link chassis QLink using SNMP.

The driver is capable to monitor/control up to 8 optical modules

It was tested with both optical transmitter and receiver modules running on SW version:

- FW: 3.1.A3.0.D1.1.S2.H1.1

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
info.odbTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.idcTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
slot.01.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.01.note	TEXT		StringRange DIS
slot.01.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.01.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.01.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.01.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.01.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.01.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.installed	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.02.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.02.note	TEXT		StringRange DIS
slot.02.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.02.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.02.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.02.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.02.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.02.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.03.note	TEXT		StringRange DIS
slot.03.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.03.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.03.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.03.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.03.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.03.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.04.note	TEXT		StringRange DIS

name	*type*	*flags*	*range*
slot.04.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.04.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.04.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.04.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.04.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.04.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.05.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.05.note	TEXT		StringRange DIS
slot.05.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.05.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.05.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.05.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.05.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.05.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.06.note	TEXT		StringRange DIS
slot.06.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.gain	INTEGER		IntegerRange DIS (0 .. 31)

name	*type*	*flags*	*range*
slot.06.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.06.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.06.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.06.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.06.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.07.note	TEXT		StringRange DIS
slot.07.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.07.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.07.threshold	INTEGER		IntegerRange DIS (-99 .. 99)
slot.07.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.07.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.07.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.card	CHOICE	R/O	EnumRange R/O (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.08.note	TEXT		StringRange DIS
slot.08.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.08.thresholdEnable	CHOICE		EnumRange DIS (ON OFF)
slot.08.threshold	INTEGER		IntegerRange DIS (-99 .. 99)

name	*type*	*flags*	*range*
slot.08.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.08.lnbNominal	INTEGER		IntegerRange DIS (0 .. 999)
slot.08.lnbWindow	INTEGER		IntegerRange DIS (0 .. 999)
slot.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 1 RF Out Overflow)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 2 RF Out Overflow)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 3 RF Out Overflow)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 4 RF Out Overflow)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 5 RF Out Overflow)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 6 RF Out Overflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 7 RF Out Overflow)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 8 RF Out Overflow)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 1 Threshold)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 2 Threshold)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 3 Threshold)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 4 Threshold)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 5 Threshold)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 6 Threshold)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 7 Threshold)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 8 Threshold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 1 LNB)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 2 LNB)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 3 LNB)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 4 LNB)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 5 LNB)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 6 LNB)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 7 LNB)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 8 LNB)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 1 TX Laser)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 2 TX Laser)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 3 TX Laser)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 4 TX Laser)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 5 TX Laser)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 6 TX Laser)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 7 TX Laser)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 8 TX Laser)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Slot 1 RX Link)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Slot 2 RX Link)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Slot 3 RX Link)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Slot 4 RX Link)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Slot 5 RX Link)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Slot 6 RX Link)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Slot 7 RX Link)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Slot 8 RX Link)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.42	ALARM	R/O	AlarmFlagRange R/O (IDC PSU 1)
faults.43	ALARM	R/O	AlarmFlagRange R/O (IDC PSU 2)
faults.44	ALARM	R/O	AlarmFlagRange R/O (ODB PSU 1)

name	*type*	*flags*	*range*
faults.45	ALARM	R/O	AlarmFlagRange R/O (ODB PSU 2)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Link 10MHz)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Signal 10MHz)
faults.48	ALARM	R/O	AlarmFlagRange R/O (IDC/ODB Communication)
faults.49	ALARM	R/O	AlarmFlagRange R/O (ODB Summary)
faults.50	ALARM	R/O	AlarmFlagRange R/O (IDC Summary)

9.550 RFDesign-RLC

This page describes the device driver and the device window for the RF Design RLC Series Combiner via SNMP.

The driver was tested with the following Models and SW Releases:

- RLC 2708L and RLC 2716L
- 2.3.26.D16.S27.H20.21.14.4

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Measurments and settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the combiner.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port

number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position.1	CHOICE		EnumRange (A B N/A)
position.2	CHOICE		EnumRange (A B N/A)
gain.1	INTEGER		IntegerRange (0 .. 25)
gain.2	INTEGER		IntegerRange (0 .. 25)
eq.1	INTEGER		IntegerRange (0 .. 9)
eq.2	INTEGER		IntegerRange (0 .. 9)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meas.input.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
agc.1	CHOICE		EnumRange (ON OFF)
agc.2	CHOICE		EnumRange (ON OFF)
agc.refLevel.1	INTEGER		IntegerRange (-40 .. 10)
agc.refLevel.2	INTEGER		IntegerRange (-40 .. 10)
agc.hysteresis.1	INTEGER		IntegerRange (1 .. 9)
agc.hysteresis.2	INTEGER		IntegerRange (1 .. 9)
agc.timeWindow.1	INTEGER		IntegerRange (1 .. 9)
agc.timeWindow.2	INTEGER		IntegerRange (1 .. 9)
threshold.1	CHOICE		EnumRange (ON OFF)
threshold.2	CHOICE		EnumRange (ON OFF)
thresholdLevel.1	INTEGER		IntegerRange (-40 .. 10)
thresholdLevel.2	INTEGER		IntegerRange (-40 .. 10)
limiter.1	CHOICE		EnumRange (ON OFF)
limiter.2	CHOICE		EnumRange (ON OFF)
limiterLevel.1	INTEGER	R/O	IntegerRange R/O (-30 .. 10)
limiterLevel.2	INTEGER	R/O	IntegerRange R/O (-30 .. 10)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 1 status)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 2 status)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Mod 1 RF pwr overflow)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Mod 2 RF pwr overflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Limiter)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Limiter)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Threshold)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Threshold)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Lower limit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Lower limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Upper limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Upper limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Lock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Amp 1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Amp 2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Amp 1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Amp 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Unit status)

9.551 RFDesign-RLS

This page describes the device driver and the device window for the RF Design RLC Series Splitter via SNMP.

The driver was tested with the following Models and SW Releases:

- RLCS716L
  - 2.3.26.D16.S27.H20.21.14.4 up to
  - 2.3.28.D16.S28.H21.23.17.4

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Measuremnts and settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the splitter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
position.1	CHOICE		EnumRange (A B N/A)
position.2	CHOICE		EnumRange (A B N/A)
gain.1	INTEGER		IntegerRange (0 .. 25)
gain.2	INTEGER		IntegerRange (0 .. 25)
eq.1	INTEGER		IntegerRange (0 .. 9)
eq.2	INTEGER		IntegerRange (0 .. 9)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
agc.1	CHOICE		EnumRange (ON OFF)
agc.2	CHOICE		EnumRange (ON OFF)
agc.refLevel.1	INTEGER		IntegerRange (-40 .. 10)
agc.refLevel.2	INTEGER		IntegerRange (-40 .. 10)
agc.hysteresis.1	INTEGER		IntegerRange (1 .. 9)
agc.hysteresis.2	INTEGER		IntegerRange (1 .. 9)
agc.timeWindow.1	INTEGER		IntegerRange (1 .. 9)
agc.timeWindow.2	INTEGER		IntegerRange (1 .. 9)
threshold.1	CHOICE		EnumRange (ON OFF)
threshold.2	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
thresholdLevel.1	INTEGER		IntegerRange (-40 .. 10)
thresholdLevel.2	INTEGER		IntegerRange (-40 .. 10)
limiter.1	CHOICE		EnumRange (ON OFF)
limiter.2	CHOICE		EnumRange (ON OFF)
limiterLevel.1	INTEGER	R/O	IntegerRange R/O (-30 .. 10)
limiterLevel.2	INTEGER	R/O	IntegerRange R/O (-30 .. 10)
meas.lnbPower.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.lnbPower.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lnb.power.1	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22kHz 18V+22kHz)
lnb.power.2	CHOICE		EnumRange (OFF 13V 15V 18V 13V+22kHz 18V+22kHz)
lnb.refLevel.1	INTEGER		IntegerRange (50 .. 350)
lnb.refLevel.2	INTEGER		IntegerRange (50 .. 350)
lnb.hysteresis.1	INTEGER		IntegerRange (10 .. 50)
lnb.hysteresis.2	INTEGER		IntegerRange (10 .. 50)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 1 status)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 2 status)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Mod 1 RF pwr overflow)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Mod 2 RF pwr overflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Limiter)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Limiter)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Threshold)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Threshold)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Lower limit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Lower limit)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Upper limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Upper limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Mod 1 AGC Lock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Mod 2 AGC Lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Amp 1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Mod 1 Amp 2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Amp 1)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Mod 2 Amp 2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Mod 1 LNB power)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Mod 2 LNB power)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Unit status)

9.552 RFDesign-RSC

This page describes the device driver and the device window for the RFDesign FlexLink-RSC201 2 to 1 RF Switch Unit.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The monitoring parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
info.chassisType	TEXT	R/O	StringRange R/O
info.moduleName	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.thresholdLevel	INTEGER	SETUP	IntegerRange (-40 .. 30)

name	*type*	*flags*	*range*
meas.inputPower.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.inputPower.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (1 2)
gain	INTEGER		IntegerRange DIS (0 .. 30)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit status)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Module status)
faults.06	ALARM	R/O	AlarmFlagRange R/O (RF power overflow)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Threshold in Slot)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Amplifier 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Amplifier 2)

9.553 RFDesign-RedLink

This page describes the device driver and the device window for the RFDesign optical link chassis RedLink.

The driver is capable to monitor/control up to 16 redundant pairs of optical transmitter modules

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	description
address	The SNMP address specification for the device (details below).

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.chassisType	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.unitName	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.fwVersion	TEXT	R/O	StringRange R/O
info.dispVersion	TEXT	R/O	StringRange R/O
internal.modStat	TEXT	R/O	StringRange R/O
slot.01.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.01.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.01.note	TEXT		StringRange DIS
slot.01.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.01.agc	CHOICE		EnumRange DIS (OFF ON)
slot.01.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.01.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.01.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.01.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.02.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.02.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.02.note	TEXT		StringRange DIS

name	*type*	*flags*	*range*
slot.02.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.02.agc	CHOICE		EnumRange DIS (OFF ON)
slot.02.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.02.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.02.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.02.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.03.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.03.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.03.note	TEXT		StringRange DIS
slot.03.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.03.agc	CHOICE		EnumRange DIS (OFF ON)
slot.03.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.03.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.03.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.03.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.03.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.04.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.04.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.04.note	TEXT		StringRange DIS
slot.04.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.04.agc	CHOICE		EnumRange DIS (OFF ON)
slot.04.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.04.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.04.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.04.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.05.installed	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.05.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.05.note	TEXT		StringRange DIS
slot.05.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.05.agc	CHOICE		EnumRange DIS (OFF ON)
slot.05.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.05.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.05.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.05.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.06.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.06.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.06.note	TEXT		StringRange DIS
slot.06.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.06.agc	CHOICE		EnumRange DIS (OFF ON)
slot.06.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.06.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.06.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.06.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.07.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.07.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.07.note	TEXT		StringRange DIS
slot.07.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.07.agc	CHOICE		EnumRange DIS (OFF ON)
slot.07.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.07.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.07.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.07.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.07.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.08.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.08.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.08.note	TEXT		StringRange DIS
slot.08.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.08.agc	CHOICE		EnumRange DIS (OFF ON)
slot.08.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.08.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.08.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.08.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.09.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.09.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.09.note	TEXT		StringRange DIS
slot.09.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.09.agc	CHOICE		EnumRange DIS (OFF ON)
slot.09.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)

name	*type*	*flags*	*range*
slot.09.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.09.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.09.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.10.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.10.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.10.note	TEXT		StringRange DIS
slot.10.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.10.agc	CHOICE		EnumRange DIS (OFF ON)
slot.10.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.10.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.10.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.10.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.10.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.11.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.11.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.11.note	TEXT		StringRange DIS
slot.11.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.11.agc	CHOICE		EnumRange DIS (OFF ON)
slot.11.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.11.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.11.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.11.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.12.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.12.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)

name	*type*	*flags*	*range*
slot.12.note	TEXT		StringRange DIS
slot.12.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.12.agc	CHOICE		EnumRange DIS (OFF ON)
slot.12.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.12.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.12.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.12.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.13.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.13.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.13.note	TEXT		StringRange DIS
slot.13.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.13.agc	CHOICE		EnumRange DIS (OFF ON)
slot.13.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.13.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.13.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.13.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.13.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.14.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.14.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.14.note	TEXT		StringRange DIS
slot.14.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.14.agc	CHOICE		EnumRange DIS (OFF ON)
slot.14.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.14.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.14.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.14.activeAmp	CHOICE		EnumRange DIS (0 1 2)

name	*type*	*flags*	*range*
slot.15.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.15.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.15.note	TEXT		StringRange DIS
slot.15.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.15.agc	CHOICE		EnumRange DIS (OFF ON)
slot.15.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.15.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.15.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.15.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.16.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.16.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.16.note	TEXT		StringRange DIS
slot.16.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.16.agc	CHOICE		EnumRange DIS (OFF ON)
slot.16.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.16.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.16.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.16.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.16.activeAmp	CHOICE		EnumRange DIS (0 1 2)
slot.17.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.17.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.17.note	TEXT		StringRange DIS
slot.17.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.17.agc	CHOICE		EnumRange DIS (OFF ON)
slot.17.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.17.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.17.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.17.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.17.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.17.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.17.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.18.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.18.note	TEXT		StringRange DIS
slot.18.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.18.agc	CHOICE		EnumRange DIS (OFF ON)
slot.18.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.18.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.18.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.18.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.19.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.19.note	TEXT		StringRange DIS
slot.19.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.19.agc	CHOICE		EnumRange DIS (OFF ON)
slot.19.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.19.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.19.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.19.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.19.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.19.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.20.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.20.note	TEXT		StringRange DIS
slot.20.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.20.agc	CHOICE		EnumRange DIS (OFF ON)
slot.20.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.20.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.20.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.20.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.21.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.21.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.21.note	TEXT		StringRange DIS
slot.21.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.21.agc	CHOICE		EnumRange DIS (OFF ON)
slot.21.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.21.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.21.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.21.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.22.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.22.note	TEXT		StringRange DIS
slot.22.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.22.agc	CHOICE		EnumRange DIS (OFF ON)
slot.22.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.22.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.22.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.22.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.22.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.22.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.23.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.23.note	TEXT		StringRange DIS
slot.23.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.23.agc	CHOICE		EnumRange DIS (OFF ON)
slot.23.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.23.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.23.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.23.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.installed	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.24.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.24.note	TEXT		StringRange DIS
slot.24.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.24.agc	CHOICE		EnumRange DIS (OFF ON)
slot.24.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.24.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.24.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.24.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.25.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.25.note	TEXT		StringRange DIS
slot.25.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.25.agc	CHOICE		EnumRange DIS (OFF ON)
slot.25.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.25.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.25.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.25.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
slot.25.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.25.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.26.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.26.note	TEXT		StringRange DIS
slot.26.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.26.agc	CHOICE		EnumRange DIS (OFF ON)
slot.26.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.26.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.26.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.26.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.27.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)

name	*type*	*flags*	*range*
slot.27.note	TEXT		StringRange DIS
slot.27.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.27.agc	CHOICE		EnumRange DIS (OFF ON)
slot.27.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.27.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.27.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.27.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.28.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.28.note	TEXT		StringRange DIS
slot.28.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.28.agc	CHOICE		EnumRange DIS (OFF ON)
slot.28.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.28.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.28.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.28.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.28.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.28.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.28.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.29.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.29.note	TEXT		StringRange DIS
slot.29.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.29.agc	CHOICE		EnumRange DIS (OFF ON)
slot.29.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.29.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.29.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.29.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.30.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.30.note	TEXT		StringRange DIS
slot.30.gain	INTEGER		IntegerRange DIS (0 .. 31)

name	*type*	*flags*	*range*
slot.30.agc	CHOICE		EnumRange DIS (OFF ON)
slot.30.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.30.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.30.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.30.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.31.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.31.note	TEXT		StringRange DIS
slot.31.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.31.agc	CHOICE		EnumRange DIS (OFF ON)
slot.31.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.31.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.31.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)

name	*type*	*flags*	*range*
slot.31.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.31.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.installed	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.32.card	CHOICE	R/O	EnumRange R/O DIS (SPLT SPLTRED AMP AMPRED TX TXRED RX RXRED SWITCH)
slot.32.note	TEXT		StringRange DIS
slot.32.gain	INTEGER		IntegerRange DIS (0 .. 31)
slot.32.agc	CHOICE		EnumRange DIS (OFF ON)
slot.32.lnbPower	CHOICE		EnumRange DIS (OFF 13V 15V 18V 13V/22kHz 18V/22kHz)
slot.32.rfPower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.diodePower	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.lnbCurrent	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.32.rfFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.thdFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.optFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcHiFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcLoFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.agcLock	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
slot.32.lnbFault	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 01)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 02)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 03)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 04)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 05)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 06)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 07)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 08)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 09)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 10)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 11)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 12)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 13)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 14)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 15)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 16)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 17)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 18)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 19)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 20)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 21)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 22)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 23)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 24)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 25)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 26)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 27)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 28)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 29)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 30)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 31)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 32)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.554 RX-Channel-ActivationBased

The RX-Channel-ActivationBased logical device controls the basic parameters of an IRD and an input switch for the IRD which selects antenna, frequency band and polarization. RX-Channel-ActivationBased replaces the RX-Channel device in ACTIVATION-BASED workflow mode on modern M&C systems which store the channel data in a central SQL database.

Although all individual IRD parameters are directly accessible with the device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input signal for the IRD. The RX-Channel-ActivationBased device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Management device, telling the actually commanded satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this confinement at any time by selecting the "manual" satellite channel.
- The RX-Channel-Activation-Based device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the RX-Channel-ActivationBased device commands a new frequency setting to the IRD.

The device implements the ACTIVATION-BASED operation mode of theRX-Channel device. This combines the 'first select the the satellite channel' paradigm of the FILTER-BY-CHANNEL mode with an advancement of the locking mechanism of the FILTER-BY-SATELLITE mode and adds a number of additional function to this. Here the most important one is, that the device is capable to select automatically an antenna which is suitable to receive the selected satellite channel. 'Activation' in this context means to select and point an antenna for the satellite channel to receive, to connect the IRD to this signal and finally lock the resources used to prevent other instances of the RX-Channel-ActivationBased device from changing parameters which would interrupt the reception.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device. The layout of this page differs for the individual operating modes
- --- The fault page
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

The typical work flow for operating this device is as follows:

1. Select the desired satellite channel from the list.
2. Set the antenna select mode either to MANUAL, AUTO or AUTO-NO-FIXED. The latter mode skips (typically small) antennas with fixed pointing when selecting the antenna.
3. With manual antenna select mode select the antenna to be used
4. Click the "ACTIVATE" button.

With automatic antenna selection, the device searches for a suitable antenna and select this automatically. It selects the antenna with this priority (highest first):

1. A fixed pointing antenna receiving the selected satellite
2. A motor driven antenna which already is pointed to the selected satellite
3. A motor driven antenna which actually is pointed to another satellite but not in use.

If more than one antenna is available in one of these search steps, the antenna with the larger diameter is used. The AUTO-NO-FIXED mode skips the first search step.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-Channel-ActivationBased device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
configFile	The name of the configuration file, the RX-Channel-ActivationBased device reads the matrix input assignments from. You may use a common configuration file for all RX-Channel-ActivationBased devices in the system or an individual file. The file format is described at the end of this chapter.

parameter	***description***
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
nlcModeId	The message ID for setting the NLC mode.
rollOffId	The message ID for settings the roll off
modulationId	The message ID for setting the modulation mode (QPSK, 8PSK etc.)
decoderInputId	The message ID to switch the IRD's decoder input between SAT/IP/ASI
bissKeyId	The message ID for setting the BISS key (session word) at the IRD. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.
bissInjectedIdId	The message ID for setting the BISS injected ID at the IRD. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.
bissModeId	The message ID for setting the BISS mode. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the IRD.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the IRD.
srcIpAddressId	The message ID for setting the IP source or multicast address at the IRD.
srcIpPortId	The message ID for setting the IP port at the IRD.
ipFecId	The message ID for enabling / disabling the IP FEC

parameter	*description*
decPhysInterfaceId	The message ID for selecting the physical network interface number at the IRD
irdLockFaultId	The message ID for the lock fault flag of the IRD.

To make the RX-Channel-ActivationBased device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies to the lock fault if a lock fault signalling in the RX-Channel-ActivationBased device is not desired.

The configuration file format

The RX-Channel's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-Channel-ActivationBased devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	*contents*	*remark*
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters.
3	polarization	A one letter code, one of the follwing:**X** for horizontal **Y** for vertical **P** if the antenna must turn the feed in order to change the polarization
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	< or >	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-Channel-ActivationBased device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |               |
#  |   antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01   ARD     X  9750.0       < 11800.0
 02   ARD     Y  9750.0       < 11800.0
 03   ARD     X  10600.0       > 11800.0
 04   ARD     Y  10600.0       > 11800.0

 05   NDR     X  9750.0       < 11800.0
 07   NDR     X  10600.0       > 11800.0

 17   SNGR     X  9750.0       < 11800.0
 18   SNGR     Y  9750.0       < 11800.0
 19   SNGR     X  10600.0       > 11800.0
 20   SNGR     Y  10600.0       > 11800.0

 09   EBU     X  10000.0       < 11800.0
 10   EBU     Y  10000.0       < 11800.0

 11   W1      X  9750.0       < 11800.0
 12   W1      Y  9750.0       < 11800.0
 23   W1      X  10600.0       > 11800.0
 24   W1      Y  10600.0       > 11800.0

 13   ASTRA    X  9750.0       < 11800.0
 14   ASTRA    Y  9750.0       < 11800.0
 15   ASTRA    X  10600.0       > 11800.0
 16   ASTRA    Y  10600.0       > 11800.0

 21   EB_12V   Y  10600.0       > 11800.0

 22   W1_V     Y  10000.0       < 11800.0
```

The RX-Channel-ActivationBased device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.frequency.min	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.frequency.max	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.bissKeyId	TEXT	SAVE SETUP	StringRange
config.bissInjectedIdId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.bissModeId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.decPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
antenna	CHOICE		EnumRange (unknown)
satellite	CHOICE		EnumRange (unknown)
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	CHOICE		EnumRange (X Y)
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)

name	*type*	*flags*	*range*
modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
bissMode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
bissKey	TEXT		StringRange
bissInjectedId	TEXT		StringRange
decoderInput	CHOICE		EnumRange (SAT IP ASI)
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
decPhysInterface	INTEGER		IntegerRange (1 .. 4)
channel	CHOICE		EnumRange (@@@channels.m)
reset	TEXT		StringRange
antennaSelect	CHOICE		EnumRange (AUTO AUTO-NO-FIXED MANUAL)
locked	BOOLEAN		BooleanRange (true false)
reserved	CHOICE	SAVE	EnumRange (NO YES)
locked.save	TEXT	SAVE	StringRange
rfLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
ipLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
copyFrom	CHOICE		EnumRange (?)
name	TEXT		StringRange
remote.channel	TEXT		StringRange
remote.antenna	CHOICE		EnumRange (MANUAL unknown)
remote.antennaSelect	CHOICE		EnumRange (AUTO AUTO-NO-FIXED MANUAL)

name	*type*	*flags*	*range*
remote.locked	BOOLEAN		BooleanRange (true false)
remote.lockResult	TEXT	R/O	StringRange R/O
remote.name	TEXT		StringRange
remote.bissMode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
remote.bissKey	TEXT		StringRange
remote.bissInjectedId	TEXT		StringRange
local.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
local.reserved	BOOLEAN	R/O	BooleanRange R/O (true false)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
actState	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Wrong antenna)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Wrong band)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Wrong polatization)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.555 RX-Channel-ByChannel

The RX-Channel-ByChannel logical device controls the basic parameters of an IRD and an input switch for the IRD which selects antenna, frequency band and polarization. RX-Channel-ByChannel replaces the RX-Channel device in FILTER-BY-CHANNEL workflow mode on modern M&C systems which store the channel data in a central SQL database. The device complements the TX-Channel-ByChannel logical device which provides a similar function for transmit chains.

Although all individual IRD parameters are directly accessible with the device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input

signal for the IRD. The RX-Channel-ByChannel device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Pointing device, telling the actually commanded satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this confinement at any time by selecting the "manual" satellite channel.
- The RX-Channel-ByChannel device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the RX-Channel-ByChannel device commands a new frequency setting to the IRD.

The FILTER-BY-CHANNEL mode is the basic operation mode of the device. it resembles the behaviour of the former RX-ChannelSelect-CDB device. This mode assumes a work flow where the operator first selects a channel from the list of stored satellite channels, then selects the input signal for the IRD. The RX-Channel-ByChannel device assists the operator with a filter / autocompletion function in the channel field (typing a partial channel name reduces the list of displayed channel to those which contain the typed character sequence) and by reducing the list of selectable IRD input signals to those signals which currently match the satellite, frequency band and polarization plane of the selected channel.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device. The layout of this page differs for the individual operating modes
- --- The fault page
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

The RX-Channel-ByChannel device implements the FILTER-BY-CHANNEL operation mode of the RX-Channel device. This makes selecting an already defined satellite channel is a simple 2-step procedure:

1. Select the desired satellite channel from the list.

2. Select the input signal.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The RX-Channel-ByChannel device forwards these settings to the IRD. The input selection list gets confined to the inputs matching the frequency band of the entered frequency.

The device also offers an option which automatically selects the input signal from the largest fixed pointing antenna which is capable to receive the channel after the channel has been selected. Manually selecting the input signal is not necessary in this case, but still possible, e.g. to override the automatic selection for using a turnable antenna.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-Channel-ByChannel device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
configFile	The name of the configuration file, the RX-Channel-ByChannel device reads the matrix input assignments from. You may use a common configuration file for all RX-Channel-ByChannel devices in the system or an individual file. The file format is described at the end of this chapter.
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
nlcModeId	The message ID for setting the NLC mode.
rollOffId	The message ID for settings the roll of
modulationId	The message ID for setting the modulation mode (QPSK, 8PSK etc.)
decoderInputId	The message ID to switch the IRD's decoder input between SAT/IP/ASI

parameter	*description*
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the IRD.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the IRD.
srcIpAddressId	The message ID for setting the IP source or multicast address at the IRD.
srcIpPortId	The message ID for setting the IP port at the IRD.
ipFecId	The message ID for enabling / disabling the IP FEC
decPhysInterfaceId	The message ID for selecting the physical network interface number at the IRD
irdLockFaultId	The message ID for the lock fault flag of the IRD.
antennaAutoSelect	Setting this to YES enables the automatic selection of an antenna when a channel is selected. Favors fixed antennas over turnable antennas, large over smaller ones.

To make the RX-Channel-ByChannel device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies to the lock fault if a lock fault signalling in the RX-Channel-ByChannel device is not desired.

The configuration file format

The RX-Channel's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-Channel-ByChannel devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	*contents*	*remark*
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters.

column	***contents***	***remark***
3	polarization	A one letter code, one of the follwing:**X** for horizontal **Y** for vertical **P** if the antenna must turn the feed in order to change the polarization
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	**<** or **>**	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-Channel-ByChannel device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |              |
#  |  antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01   ARD     X  9750.0       < 11800.0
 02   ARD     Y  9750.0       < 11800.0
 03   ARD     X  10600.0       > 11800.0
 04   ARD     Y  10600.0       > 11800.0

 05   NDR     X  9750.0       < 11800.0
 07   NDR     X  10600.0       > 11800.0

 17   SNGR     X  9750.0       < 11800.0
 18   SNGR     Y  9750.0       < 11800.0
 19   SNGR     X  10600.0       > 11800.0
 20   SNGR     Y  10600.0       > 11800.0

 09   EBU     X  10000.0       < 11800.0
 10   EBU     Y  10000.0       < 11800.0

 11   W1      X  9750.0       < 11800.0
 12   W1      Y  9750.0       < 11800.0
 23   W1      X  10600.0       > 11800.0
 24   W1      Y  10600.0       > 11800.0

 13   ASTRA    X  9750.0       < 11800.0
 14   ASTRA    Y  9750.0       < 11800.0
 15   ASTRA    X  10600.0       > 11800.0
 16   ASTRA    Y  10600.0       > 11800.0

 21   EB_12V   Y  10600.0       > 11800.0

 22   W1_V     Y  10000.0       < 11800.0
```

The RX-Channel-ByChannel device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.frequency.min	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.frequency.max	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.decPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
config.antennaAutoSelect	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.listMode	CHOICE	SAVE SETUP	EnumRange (NAME ALIAS)
antenna	TEXT	R/O	StringRange R/O
satellite	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	CHOICE		EnumRange (X Y)
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)

name	*type*	*flags*	*range*
decoderInput	CHOICE		EnumRange (SAT IP ASI)
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
decPhysInterface	INTEGER		IntegerRange (1 .. 4)
channel	CHOICE		EnumRange (unknown)
reset	TEXT		StringRange
locked	BOOLEAN	SAVE	BooleanRange (true false)
channel.save	TEXT	SAVE NOPRESET	StringRange
rfLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
ipLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
copyFrom	CHOICE		EnumRange (?)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Wrong antenna)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Wrong band)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Wrong polatization)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.556 RX-Channel-BySatellite

The RX-Channel-BySatellite logical device controls the basic parameters of an IRD and an input switch for the IRD which selects antenna, frequency band and polarization. RX-Channel-BySatellite replaces the RX-Channel device in FILTER-BY-Satellite workflow mode on modern M&C systems which store the channel data in a central SQL database. The device complements the TX-Channel-BySatellite logical device which provides a similar function for transmit chains.

Although all individual IRD parameters are directly accessible with the device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input signal for the IRD. The RX-Channel-BySatellite device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Pointing device, telling the actually commanded satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this confinement at any time by selecting the "manual" satellite channel.
- The RX-Channel-BySatellite device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the device commands a new frequency setting to the IRD.

With this device, the operator first selects the antenna to be used, then the satellite this antenna shall be pointed to. Selecting the satellite actually commands the antenna controller to point the antenna to this satellite. Next the device reduces the list of displayed satellite channels to those which may be received with this antenna/satellite. After selecting a channel from the list, the IRD parameters are set, the IRD input signal is selected automatically and - if necessary - the antenna is moved to point to the satellite referenced in the selected channel. Other functions of this device are:

- A locking mechanism which protects an antenna from being turned away while receiving a certain signal. An operator may "lock" an IRD which prevents the operator from changing any IRD parameters in the device screen and from turning away the antenna from this or from any other RX-Channel-BySatellite or TX-Channel-BySatellite devices.
- The locking mechanism used with this mode may be configured, that a lock only may be released from the same client PC which has set the lock.
- The device is aware of antennas with 2-port feed which may receive / transmit only at one polarization plane at a time. Locking an IRD receiving from such an antenna also locks the polarization plan selection for all devices to prevent the signal for the first IRD from being interrupted.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device. The layout of this page differs for the individual operating modes
- --- The fault page
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

With this device, the 'channel' field only shows the channels of the satellite the selected antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the RX-Channel-BySatellite device screen may me used or the antenna may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. The typical sequence of operation in this mode is as follows:

1. Select the antenna to receive from (field "antenna")
2. Select the satellite to receive with this antenna if this needs to be changed (field "satellite")
3. Select the satellite channel to receive. (field "channel", shows only channels from the selected satellite)

Selecting the satellite channel will automatically select the appropriate l-band matrix input and set the IRD parameters. Only if the input selection is ambiguous, e.g. if the satellite can be received through a fixed and a motor driven antenna at the same time, the input selection must be done explicitly.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-Channel-BySatellite device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.

parameter	***description***
configFile	The name of the configuration file, the RX-Channel-BySatellite device reads the matrix input assignments from. You may use a common configuration file for all RX-Channel-BySatellite devices in the system or an individual file. The file format is described at the end of this chapter.
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
nlcModeId	The message ID for setting the NLC mode.
rollOffId	The message ID for settings the roll of
modulationId	The message ID for setting the modulation mode (QPSK, 8PSK etc.)
decoderInputId	The message ID to switch the IRD's decoder input between SAT/IP/ASI
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the IRD.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the IRD.
srcIpAddressId	The message ID for setting the IP source or multicast address at the IRD.
srcIpPortId	The message ID for setting the IP port at the IRD.
ipFecId	The message ID for enabling / disabling the IP FEC
decPhysInterfaceId	The message ID for selecting the physical network interface number at the IRD
irdLockFaultId	The message ID for the lock fault flag of the IRD.
lockByIP	Setting this to YES forces the device to require a lock to be released from the same computer from where it was set.

To make the RX-Channel-BySatellite device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies

to the lock fault if a lock fault signalling in the RX-Channel-BySatellite device is not desired.

The configuration file format

The RX-Channel's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-Channel-BySatellite devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	***contents***	***remark***
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas or pipe characters.
3	polarization	A one letter code, one of the follwing:**X** for horizontal **Y** for vertical **P** if the antenna must turn the feed in order to change the polarization
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	**<** or **>**	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-Chnnel-BySatellite device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |              |
#  |  antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01   ARD     X  9750.0       < 11800.0
 02   ARD     Y  9750.0       < 11800.0
 03   ARD     X  10600.0       > 11800.0
 04   ARD     Y  10600.0       > 11800.0

 05   NDR     X  9750.0       < 11800.0
 07   NDR     X  10600.0       > 11800.0

 17   SNGR     X  9750.0       < 11800.0
 18   SNGR     Y  9750.0       < 11800.0
 19   SNGR     X  10600.0       > 11800.0
 20   SNGR     Y  10600.0       > 11800.0

 09   EBU     X  10000.0       < 11800.0
 10   EBU     Y  10000.0       < 11800.0

 11   W1      X  9750.0       < 11800.0
 12   W1      Y  9750.0       < 11800.0
 23   W1      X  10600.0       > 11800.0
 24   W1      Y  10600.0       > 11800.0

 13   ASTRA    X  9750.0       < 11800.0
 14   ASTRA    Y  9750.0       < 11800.0
 15   ASTRA    X  10600.0       > 11800.0
 16   ASTRA    Y  10600.0       > 11800.0

 21   EB_12V   Y  10600.0       > 11800.0

 22   W1_V     Y  10000.0       < 11800.0
```

The RX-Channel-BySatellite device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.frequency.min	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.frequency.max	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.decPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.listMode	CHOICE	SAVE SETUP	EnumRange (NAME ALIAS)
antenna	CHOICE		EnumRange (unknown)
satellite	CHOICE		EnumRange (unknown)
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	CHOICE		EnumRange (X Y)
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)

name	*type*	*flags*	*range*
decoderInput	CHOICE		EnumRange (SAT IP ASI)
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
decPhysInterface	INTEGER		IntegerRange (1 .. 4)
channel	CHOICE		EnumRange (unknown)
shortChan	CHOICE		EnumRange (unknown)
reset	TEXT		StringRange
locked	BOOLEAN	R/O	BooleanRange R/O (true false)
satLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
polLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
channel.save	TEXT	SAVE NOPRESET	StringRange
rfLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
ipLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedPol	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedFreq	BOOLEAN	R/O	BooleanRange R/O (true false)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Wrong antenna)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Wrong band)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Wrong polatization)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.557 RX-Channel

The RX-Channel logical device controls the basic parameters of an IRD and an input switch for the IRD which selects antenna, frequency band and polarization. The device complements the TX-Channel logical device which provides a similar function for transmit chains.

RX-Channel supersedes a number of now outdated device like theRX-LineSettings-CDB device, which no longer should be used for new applications. The RX-Channel device supports a larger number of parameters, may be used in M&C and in NMS environments and it supports three alternative operation modes / work flows.

Heart of the RX-Channel device is a central database (CDB) storing the line settings for all satellite channels used in the system. In a NMS environment the database is located in the NMS server, providing an synchronization mechanism which creates local (read-only) copies of the database for offline operation of the VLCs. In a NMS environment, the sat-nms client software looks up the channel lists directly at the NMS database server which significantly reduces the amount of data which has to be exchanged between NMS and VLC. In a M&C environment the CDB is located at the M&C server, no synchronization is required in this case.

Although all individual IRD parameters are directly accessible with the RX-Channel device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input signal for the IRD. The RX-Channel device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Pointing device, telling the actually commanded satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this

confinement at any time by selecting the "manual" satellite channel.
- The RX-Channel device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the RX-Channel device commands a new frequency setting to the IRD.

Operation Modes / Work Flows

The RX-Channel device supports three different operation modes, thus is capable to be configured to meet the requirements of various work flows. The configuration parameter "workflowMode" selects the mode to be used.

FILTER-BY-CHANNEL The FILTER-BY-CHANNEL mode is the basic operation mode of the device. it resembles the behaviour of the former RX-ChannelSelect-CDB device. This mode assumes a work flow where the operator first selects a channel from the list of stored satellite channels, then selects the input signal for the IRD. The RX-Channel device assists the operator with a filter / autocompletion function in the channel field (typing a partial channel name reduces the list of displayed channel to those which contain the typed character sequence) and by reducing the list of selectable IRD input signals to those signals which currently match the satellite, frequency band and polarization plane of the selected channel.

FILTER-BY-SATELLITE The FILTER-BY-SATELLITE mode turns the work flow the other way round: The operator first selects the antenna to be used, then the satellite this antenna shall be pointed to. Selecting the satellite actually commands the antenna controller to point the antenna to this satellite. Next the device reduces the list of displayed satellite channels to those which may be received with this antenna/satellite. After selecting a channel from the list, the IRD parameters are set, the IRD input signal is selected automatically and - if necessary - the antenna is moved to point to the satellite referenced in the selected channel. Other functions of this mode are:

- A locking mechanism which protects an antenna from being turned away while receiving a certain signal. An operator may "lock" an IRD which prevents the operator from changing any IRD parameters in the RX-Channel device screen and from turning away the antenna from this or from any other RX-Channel or TX-Channel device.
- The locking mechanism used with this mode may be configured, that a lock only may be released from the same client PC which has set the lock.
- The device is aware of antennas with 2-port feed which may receive / transmit only at one polarization plane at a time. Locking an IRD receiving from such an antenna also locks the polarization plan selection for all devices to prevent the signal for the first IRD from being interrupted.

ACTIVATION-BASED The ACTIVATION-BASED operation mode is the most versatile of the three modes. It combines the 'first select the the satellite channel' paradigm of the FILTER-BY-CHANNEL mode with an advancement of the locking mechanism of the FILTER-BY-SATELLITE mode and adds a number of additional function to this. Here the most important one is, that the device is capable to select automatically an antenna which is suitable to receive the selected satellite channel. 'Activation' in this context means to select and point an antenna for the satellite channel to receive, to connect the IRD to this signal and finally lock the resources

used to prevent other instances of the RX-Channel device from changing parameters which would interrupt the reception.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device. The layout of this page differs for the individual operating modes
- --- The line settings database editor page.
- --- The fault page
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

The RX-Channel device supports three different operation modes / work flow settings which are selected by the 'operationMode' configuration parameter.

With 'operationMode=FILTER-BY-CHANNEL' (the default mode), selecting an already defined satellite channel is a simple 2-step procedure:

1. Select the desired satellite channel from the list.
2. Select the input signal.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The RX-Channel device forwards these settings to the IRD. The input selection list gets confined to the inputs matching the frequency band of the entered frequency.

The FILTER-BY-CHANNEL mode also offers an option which automatically selects the input signal from the largest fixed pointing antenna which is capable to receive the channel after the channel has been selected. Manually selecting the input signal is not necessary in this case, but still possible, e.g. to override the automatic selection for using a turnable antenna.

With 'operationMode=FILTER-BY-SATELLITE', the 'channel' field only shows the channels of the satellite the selected antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the TX-LineSettings-CDB's device screen may me used or the antenna may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. The typical sequence of operation in this mode is as follows:

1. Select the antenna to receive from (field "antenna")
2. Select the satellite to receive with this antenna if this needs to be changed (field

"satellite")
3. Select the satellite channel to receive. (field "channel", shows only channels from the selected satellite) Selecting the satellite channel will automatically select the appropriate l-band matrix input and set the IRD parameters. Only if the input selection is ambiguous, e.g. if the satellite can be received through a fixed and a motor driven antenna at the same time, the input selection must be done explicitly.

With 'operationMode=ACTIVATION-BASED' the typical work flow is as follows:

1. Select the desired satellite channel from the list.
2. Set the antenna select mode either to MANUAL, AUTO or AUTO-NO-FIXED. The latter mode skips (typically small) antennas with fixed pointing when selecting the antenna.
3. With manual antenna select mode select the antenna to be used
4. Click the "ACTIVATE" button.

With automatic antenna selection, the device searches for a suitable antenna and select this automatically. It selects the antenna with this priority (highest first):

1. A fixed pointing antenna receiving the selected satellite
2. A motor driven antenna which already is pointed to the selected satellite
3. A motor driven antenna which actually is pointed to another satellite but not in use. If more than one antenna is available in one of these search steps, the antenna with the larger diameter is used. The AUTO-NO-FIXED mode skips the first search step.

Operation (database edit)

With the second page of the device window of the RX-Channel device channel definitions are edited, copied or deleted. All RX-Channel devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the RX-Channel device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete

2. Click 'DELETE'

Local database copy

The following paragraph only applies to NMS installations. M&C installation manage their CDB locally at the M&C server and do not require a local copy of the data.

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the RX-Channel functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a RX-LineSettings-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the RX-Channel device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS

server, the RX-Channel device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-LineSettings-CDB device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
configFile	The name of the configuration file, the RX-LineSettings-CDB device reads the matrix input assignments from. You may use a common configuration file for all RX-LineSettings-CDB devices in the system or an individual file. The file format is described at the end of this chapter.
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
rollOffId	The message ID for settings the roll of
modulationId	The message ID for setting the modulation mode (QPSK, 8PSK etc.)
decoderInputId	The message ID to switch the IRD's decoder input between SAT/IP/ASI
bissKeyId	The message ID for setting the BISS key (session word) at the IRD. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.

parameter	***description***
bissInjectedIdId	The message ID for setting the BISS injected ID at the IRD. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.
bissModeId	The message ID for setting the BISS mode. Please note, that this parameter is only used in ACTIVATION-BASED mode and should only be set if you do not intend to use the sat-nms BISS key CDB device for setting the encryption parameters of the IRD.
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the IRD.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the IRD.
srcIpAddressId	The message ID for setting the IP source or multicast address at the IRD.
srcIpPortId	The message ID for setting the IP port at the IRD.
ipFecId	The message ID for enabling / disabling the IP FEC
decPhysInterfaceId	The message ID for selecting the physical network interface number at the IRD
irdLockFaultId	The message ID for the lock fault flag of the IRD.
operationMode	Selects the workflow / operation mode of the device. Operation modes FILTER-BY-SATELLITE and ACTIVATION-BASED are only possible in M&C environments.
lockByIP	Setting this to YES forces the device in FILTER-BY-SATELLITE mode to require a lock to be released from the same computer from where it was set. With the other operation modes 'lockByIP' is ignored.
antennaAutoSelect	Setting this to YES enables the automatic selection of an antenna when a channel is selected. Favors fixed antennas over turnable antennas, large over smaller ones. This parameter is only available in FILTER-BY-CHANNEL mode, in other modes the parameter appears grayed out.

To make the RX-Channel device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies to the lock fault if a lock fault signalling in the RX-Channel device is not desired.

The configuration file format

The RX-Channel's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-Channel devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	***contents***	***remark***
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters ('
3	polarization	A one letter code, one of the follwing:**X** for horizontal **Y** for vertical **P** if the antenna must turn the feed in order to change the polarization
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	**<** or **>**	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-LineSettings-CDB device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |              |
#  |  antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01   ARD     X  9750.0       < 11800.0
 02   ARD     Y  9750.0       < 11800.0
 03   ARD     X  10600.0       > 11800.0
 04   ARD     Y  10600.0       > 11800.0

 05   NDR     X  9750.0       < 11800.0
 07   NDR     X  10600.0       > 11800.0

 17   SNGR     X  9750.0       < 11800.0
 18   SNGR     Y  9750.0       < 11800.0
 19   SNGR     X  10600.0       > 11800.0
 20   SNGR     Y  10600.0       > 11800.0

 09   EBU     X  10000.0       < 11800.0
 10   EBU     Y  10000.0       < 11800.0

 11   W1      X  9750.0       < 11800.0
 12   W1      Y  9750.0       < 11800.0
 23   W1      X  10600.0       > 11800.0
 24   W1      Y  10600.0       > 11800.0

 13   ASTRA    X  9750.0       < 11800.0
 14   ASTRA    Y  9750.0       < 11800.0
 15   ASTRA    X  10600.0       > 11800.0
 16   ASTRA    Y  10600.0       > 11800.0

 21   EB_12V   Y  10600.0       > 11800.0

 22   W1_V     Y  10000.0       < 11800.0
```

The RX-LineSettings-CDB device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.bissKeyId	TEXT	SAVE SETUP	StringRange
config.bissInjectedIdId	TEXT	SAVE SETUP	StringRange
config.bissModeId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ipProtocolId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.decPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
config.workflowMode	CHOICE	SAVE SETUP	EnumRange (FILTER-BY-CHANNEL FILTER-BY-SATELLITE ACTIVATION-BASED DB-ONLY)
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.antennaAutoSelect	CHOICE	SAVE SETUP	EnumRange (NO YES)
antenna	CHOICE		EnumRange (unknown)
satellite	CHOICE		EnumRange (unknown)
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	CHOICE		EnumRange (X Y)
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)

name	*type*	*flags*	*range*
nlcMode	CHOICE		EnumRange (OFF ON)
rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
bissMode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
bissKey	TEXT		StringRange
bissInjectedId	TEXT		StringRange
decoderInput	CHOICE		EnumRange (SAT IP ASI)
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
decPhysInterface	INTEGER		IntegerRange (1 .. 4)
channel	CHOICE		EnumRange (@@@channels.m)
shortChan	CHOICE		EnumRange (unknown)
reset	TEXT		StringRange
antennaSelect	CHOICE		EnumRange (AUTO AUTO-NO-FIXED MANUAL)
locked	BOOLEAN		BooleanRange (true false)
reserved	CHOICE	SAVE	EnumRange (NO YES)
locked.save	TEXT	SAVE	StringRange
satLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
polLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
togglelock	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
locksource	TEXT	R/O	StringRange R/O
channel.save	TEXT	SAVE NOPRESET	StringRange
rfLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
ipLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedPol	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedFreq	BOOLEAN	R/O	BooleanRange R/O (true false)
copyFrom	CHOICE		EnumRange (?)
name	TEXT		StringRange
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	CHOICE		EnumRange (NONE)
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y L R)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y L R)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.bitRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.useSymbolRate	CHOICE		EnumRange (SYMBOLRATE BITRATE)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
edit.packetSize	CHOICE		EnumRange (188 204)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)

name	*type*	*flags*	*range*
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.nlcMode	CHOICE		EnumRange (OFF ON)
edit.frameSize	CHOICE		EnumRange (SHORT NORMAL)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.videoRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.videoRateAuto	CHOICE		EnumRange (FIXED AUTO)
edit.profile	CHOICE		EnumRange (MPEG2-MP@ML-420 MPEG2-MP@LL-420 MPEG2-HP@ML-422 MPEG2-HP@LL-422 MPEG2-SP@LL-420 MPEG2-SP@ML-420 MPEG2-MP@H14-420 MPEG2-MP@HL-420 MPEG2-HP@H14-422 MPEG2-HP@HL-422 MPEG2-HP@HL-420 MPEG2-HP@H14-420 MPEG2-HP@ML-420 MPEG4-BP@L2-420 MPEG4-BP@L3-420 MPEG4-BP@L4-420 MPEG4-MP@L3-420 MPEG4-MP@L4-420 MPEG4-MP@L4.2-420 MPEG4-HP@L3-420 MPEG4-HP@L4-420 MPEG4-HP@L4.2-420 MPEG4-HP@L3-422 MPEG4-HP@L4-422 MPEG4-HP@L4.2-422)
edit.eirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.reducedEirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.autoLineUp	CHOICE		EnumRange DIS (OFF ON)
edit.audioRate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio2Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)

name	*type*	*flags*	*range*
edit.audio3Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio4Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audioChannels	CHOICE		EnumRange (1 2 3 4)
edit.comment	TEXT		StringRange
edit.user	TEXT		StringRange
edit.programTitle	TEXT		StringRange
edit.decoderInput	CHOICE		EnumRange (SAT IP ASI)
edit.ipType	CHOICE		EnumRange (UNICAST MULTICAST)
edit.ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
edit.srcIpAddress	TEXT		StringRange
edit.srcIpPort	INTEGER		IntegerRange (0 .. 65535)
edit.dstIpAddress	TEXT		StringRange
edit.dstIpPort	INTEGER		IntegerRange (0 .. 65535)
edit.ipFec	CHOICE		EnumRange (OFF ON)
edit.ipBufferSize	INTEGER		IntegerRange (0 .. 0)
edit.encPhysInterface	INTEGER		IntegerRange (1 .. 4)
edit.decPhysInterface	INTEGER		IntegerRange (1 .. 4)
edit.satEnable	TEXT	R/O	StringRange R/O
edit.ipEnable	TEXT	R/O	StringRange R/O
edit.brEnable	TEXT	R/O	StringRange R/O
edit.srEnable	TEXT	R/O	StringRange R/O
edit.vrEnable	TEXT	R/O	StringRange R/O
edit.ar2Enable	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
edit.ar3Enable	TEXT	R/O	StringRange R/O
edit.ar4Enable	TEXT	R/O	StringRange R/O
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
remote.channel	TEXT		StringRange
remote.antenna	CHOICE		EnumRange (MANUAL unknown)
remote.antennaSelect	CHOICE		EnumRange (AUTO AUTO-NO-FIXED MANUAL)
remote.locked	BOOLEAN		BooleanRange (true false)
remote.lockResult	TEXT	R/O	StringRange R/O
remote.name	TEXT		StringRange
remote.bissMode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E)
remote.bissKey	TEXT		StringRange
remote.bissInjectedId	TEXT		StringRange
local.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
local.reserved	BOOLEAN	R/O	BooleanRange R/O (true false)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
actState	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Wrong antenna)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Wrong band)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Wrong polatization)

9.558 RX-ChannelSelect-CDB-DBonly

The RX-ChannelSelect-CDB logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization.

This is a special version of the device, which provides only access to the database.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The channel database editor page.
- --- The fault page.
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (database edit)

With the second page of the device window of the RX-ChannelSelect-CDB device channel definitions are edited, copied or deleted. All RX-ChannelSelect devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the RX-ChannelSelect-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some

punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.

- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete
  2. Click 'DELETE'

Local database copy

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the RX-ChannelSelect-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a RX-ChannelSelect-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the RX-ChannelSelect-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to

fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the RX-ChannelSelect-CDB device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

Just leave the configuration parameters empty.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
antenna	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange (unknown)

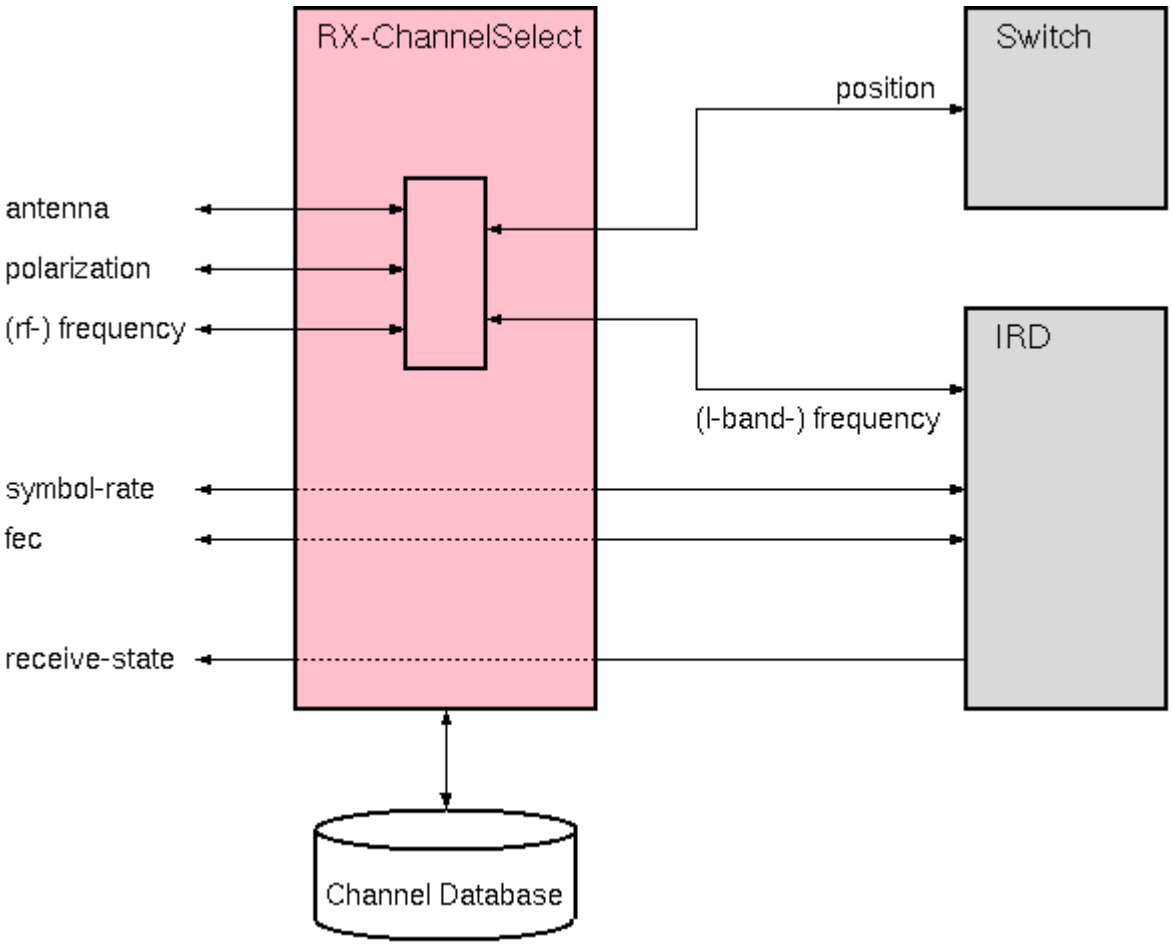
name	*type*	*flags*	*range*
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	TEXT	R/O	StringRange R/O
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
channel	CHOICE		EnumRange (@@@channels.m)
reset	TEXT		StringRange
locked	BOOLEAN	SAVE	BooleanRange (true false)
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.nlcMode	CHOICE		EnumRange (OFF ON)
edit.pilots	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
edit.comment	TEXT		StringRange
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

9.559 RX-ChannelSelect-CDB

The RX-ChannelSelect-CDB logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization. RX-ChannelSelect-CDB is comparable to the RX-ChannelSelect device, but is intendended for the use in a sat-nms NMS system. "CDB" in the driver name stands for "Central Data Base", underlining that all RX-ChannelSelect-CDB instances use one central satellite channel database located at the NMS server. The RX-ChannelSelect-CDB device primarily is used for VLC configurations where several IRDs are connected to an L-band switch matrix which distributes the signals from multiple antennas. It however can be used for simpler applications as well.

The RX-ChannelSelect-CDB device is complemented by theTX-ChannelSelect-CDB logical device which may be used to control the equipment of a (DVB) transmit chain in a similar way. Both ChannelSelect devices share the same channel database.



Although all individual IRD parameters are individually accessible with the RX-ChannelSelect-CDB device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input signal for the IRD. The RX-ChannelSelect-CDB device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Pointing device, telling the actually commanded satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this confinement at any time by selecting the "manual" satellite channel.
- The RX-ChannelSelect-CDB device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the RX-ChannelSelect-CDB device commands a new frequency setting to the IRD.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The channel database editor page.
- --- The fault page.
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

Selecting an already defined satellite channel is a simple 2-step procedure:

1. Select the desired satellite channel from the list.
2. Select the input signal.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The RX-ChannelSelect-CDB device forwards these settings to the IRD. The input selection list gets confined to the inputs matching the frequency band of the entered frequency.

Operation (database edit)

With the second page of the device window of the RX-ChannelSelect-CDB device channel definitions are edited, copied or deleted. All RX-ChannelSelect devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the RX-ChannelSelect-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'

- Delete a channel:
    1. Select the channel to delete
    2. Click 'DELETE'

Local database copy

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the RX-ChannelSelect-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a RX-ChannelSelect-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the RX-ChannelSelect-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the RX-ChannelSelect-CDB device is not involved in this and does not recognize the

new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-ChannelSelect-CDB device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
configFile	The name of the configuration file, the RX-ChannelSelect-CDB device reads the matrix input assignments from. You may use a common configuration file for all RX-ChannelSelect-CDB devices in the system or an individual file. The file format is described at the end of this chapter.
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
irdLockFaultId	The message ID for the lock fault flag of the IRD.

To make the RX-ChannelSelect-CDB device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies to the lock fault if a lock fault signalling in the RX-ChannelSelect-CDB device is not desired.

The configuration file format

The RX-ChannelSelect's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-ChannelSelect-CDB devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	*contents*	*remark*
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters ('
3	polarization	A one letter code, one of the following: **X** for horizontal **Y** for vertical
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	< or >	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-ChannelSelect-CDB device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |              |
#  |  antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01   ARD     X  9750.0       < 11800.0
 02   ARD     Y  9750.0       < 11800.0
 03   ARD     X  10600.0       > 11800.0
 04   ARD     Y  10600.0       > 11800.0

 05   NDR     X  9750.0       < 11800.0
 07   NDR     X  10600.0       > 11800.0

 17   SNGR     X  9750.0       < 11800.0
 18   SNGR     Y  9750.0       < 11800.0
 19   SNGR     X  10600.0       > 11800.0
 20   SNGR     Y  10600.0       > 11800.0

 09   EBU     X  10000.0       < 11800.0
 10   EBU     Y  10000.0       < 11800.0

 11   W1      X  9750.0       < 11800.0
 12   W1      Y  9750.0       < 11800.0
 23   W1      X  10600.0       > 11800.0
 24   W1      Y  10600.0       > 11800.0

 13   ASTRA    X  9750.0       < 11800.0
 14   ASTRA    Y  9750.0       < 11800.0
 15   ASTRA    X  10600.0       > 11800.0
 16   ASTRA    Y  10600.0       > 11800.0

 21   EB_12V   Y  10600.0       > 11800.0

 22   W1_V     Y  10000.0       < 11800.0
```

The RX-ChannelSelect-CDB device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

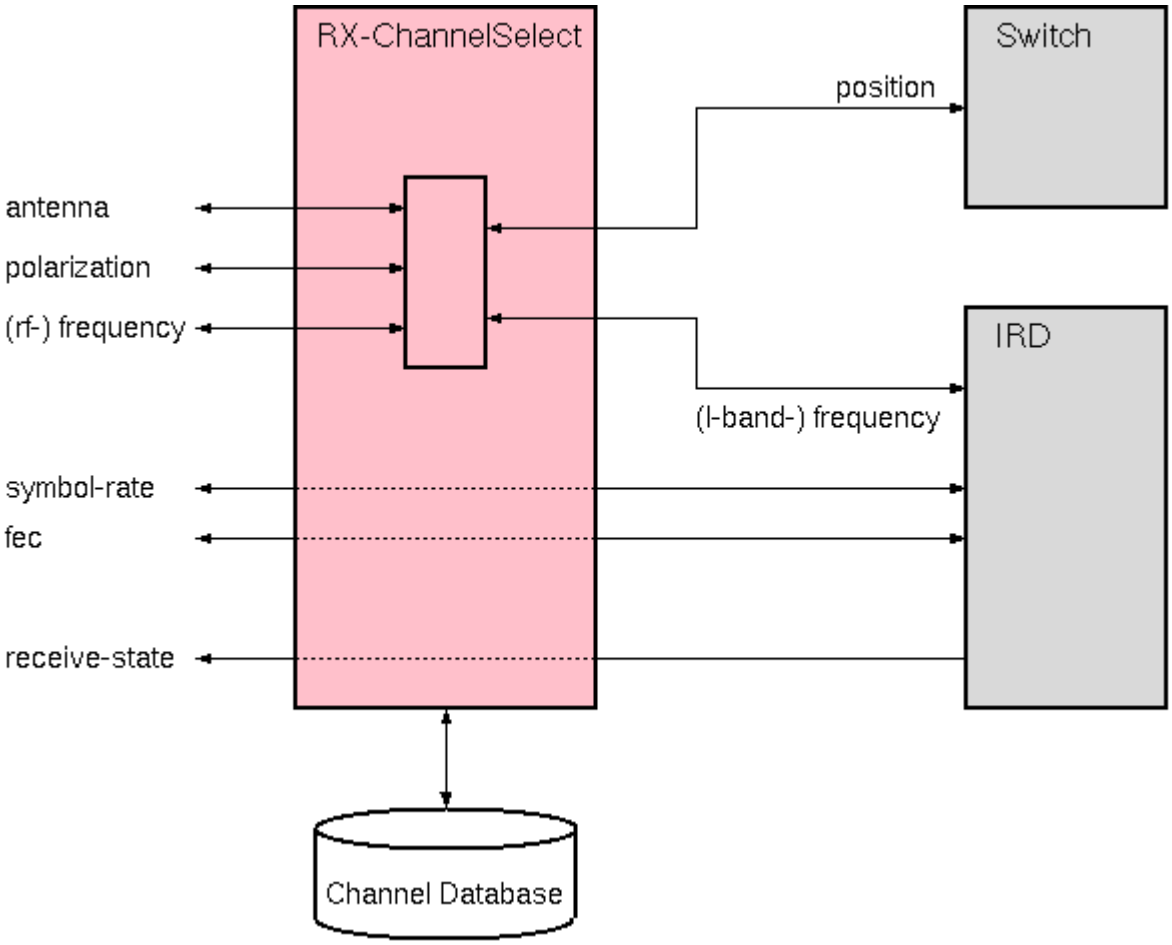
name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
antenna	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	TEXT	R/O	StringRange R/O
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
channel	CHOICE		EnumRange (@@@channels.m)

name	*type*	*flags*	*range*
reset	TEXT		StringRange
locked	BOOLEAN	SAVE	BooleanRange (true false)
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.nlcMode	CHOICE		EnumRange (OFF ON)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.comment	TEXT		StringRange
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Wrong input)

9.560 RX-ChannelSelect

The RX-ChannelSelect logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization. RX-ChannelSelect supersedes IRD-Chain device, adding new features like a system wide common satellite channel database. The RX-ChannelSelect device primarily is intended for configurations where several IRDs are connected to an L-band switch matrix which distributes the signals from multiple antennas. It however can be used for simpler applications as well.



The RX-ChannelSelect device translates the user level parameters 'antenna', 'satellite', 'polarization' and 'frequency' into the frequency setting at the IRD and the position of the input selection switch / matrix output. The translation is made following the rules defined in a configuration file. This file describes the switch matrix inputs. The RX-ChannelSelect device does the translation of LO-frequencies, as these may vary for the different antennas.

Beside the function described above, the RX-ChannelSelect device implements a locking mechanism which prevents users from changing parameters at a receive chain which is currently in use. This protection only works on the parameters controlled by the RX-ChannelSelect device, it does not influence the operation of the IRD through it's own screen.

The RX-ChannelSelect device is complemented by theTX-ChannelSelect logical device which may be used to control the equipment of a (DVB) transmit chain in a similar way. Both ChannelSelect devices share the same channel database and the same locking mechanism: While on RX-ChannelSelect device locks an antenna pointing, the TX-ChannelSelect device using the same antenna cannot move this locked antenna.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The channel database editor page.
- --- The fault page
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
  - **IRD Summary** : The associated IRD shows a summary fault (includes Lock fault).
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

Selecting an already defined satellite channels is a simple 3-step procedure:

1. Select the antenna you want to use at the 'antenna' drop down selection box. This fills the 'satellite' box with a list of satellites (antenna pointing) stored with this antenna. The satellite the antenna actually is pointed to is pre-selected.
2. Now select the satellite you want to receive. The antenna gets commanded to point to the selected satellite and the 'channel' selection gets filled with the channel selection stored for this satellite.
3. Select the channel you want to receive. This tunes IRD to the frequency / symbol rate / fec settings stored for this channel and commands the l-band matrix output port to select the antenna port associated with this frequency abd / polarization.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The RX-ChannelSelect device forwards these settings to the IRD and selects the l-band matrix input to route the selected signal from the antenna to the IRD.

Operation (database edit)

With the second page of the device window of the RX-ChannelSelect device channel definitions

are edited, copied or deleted. All RX-ChannelSelect devices in a sat-nms M&C share one common channel database. Changes you make at one device apply to all others as well. Selecting channels at his page or changing any channel parameters do not change the receiver settings.

To edit the channel database, first select the satellite which broadcasts the channel you want to edit. The 'satellite' selection offers all satellites, receivable by any antenna of the station as a whole. The subsequent steps depend on what you want to do. Here a few examples:

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name, this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.
- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit frequency / polarization.
  3. Enter a new (for this satellite unique) channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete
  2. Click 'DELETE'

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-ChannelSelect device with the IRD and the input switch.

parameter	***description***
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
lockByIP	If you set this parameter to 'YES', only the operator who 'locked' the RX-ChannelSelect may unlock it again. He is required to do this from the same workplace where he locked the device. When set to 'NO' any operator logged in with privilege level 100 or more may unlock a locked RX-ChannelSelect from any workplace.
configFile	The name of the configuration file, the RX-ChannelSelect reads the matrix input assignments from. You may use a common configuration file for all RX-ChannelSelect devices in the system or an individual file. The file format is described at the end of this chapter.

parameter	*description*
refChannel	The name of the reference channel used to search and identify a satellite with mobile SNG applications. See section "Reference channels" below for a detailed discussion of this.

Reference channels

For mobile SNG applications the RX-ChannelSelect device is capable to tune the IRD to a special channel as soon as the antenna gets moved to a satellite. This special channel is used to search and identify the satellite. To setup the RX-ChannelSelect device to use this feature, configure the following settings:

At the maintenance page of the RX-ChannelSelect device set the 'refChannel' configuration setting to a non-empty value. For most applications, it makes sense to use the same channel name for all instances of the device. Usually you will name the reference channel "Find-Satellite", "default" or something similar. You also may want to prepend the name with a "Z" to get the reference channel sorted at the very end of the list "Z-default".

Now, for each satellite of interest, define a channel with the name you defined above. The channel should address a DVB signal at the satellite which is very likely to be present all the time.

If you select a satellite with the reference channel setup made as described above, the RX-ChannelSelect device commands the satellite name to the antenna and in parallel it tunes the IRD to the parameters of the reference channel.

If for the given satellite no reference channel is available or if the reference channel name isn't configured at all, the RX-ChannelSelect device will not program the receiver until the operator explicitly selects a channel.

Configuring an M&C with RX-ChannelSelect devices

Unlike other logical devices, which know all links to other parameters / devices from configuration settings, the RX-ChannelSelect device refers to the IRD it controls by it's name. This mainly is because of the IRD device icon which is contained in the RX-ChannelSelect's device window.

The name of the IRD controlled by an RX-ChannelSelect device always must start with an "I" followed by the suffix contained in the RX-ChannelSelect name. The suffix is everything including and following the last hyphen in the name or any trailing digits. The list below shows a few examples for explanation:

RX-ChannelSelect name	*IRD name*
CHAIN1	I1
CHN-SCOPUS4	I-SCOPUS4

RX-ChannelSelect name	*IRD name*
CH-RCV-T3	I-T3
CHAIN	ICHAIN

When you select a name for the RX-ChannelSelect device, you **must** name the IRD following this scheme to make the IRD being controlled by the RX-ChannelSelect.

Using the lock/unlock mechanism in a task-oriented screen

The RX-ChannelSelect's lock/unlock mechanism works through the variables 'locked' and 'togglelock'.

To make a parameter in a task-oriented screen 'protected' by the RX-ChannelSelect device (i.e. the parameter only can be changed it the RX-ChannelSelect is 'unlocked'), set the 'locked' flag of the RX-ChannelSelect as the 'enable ID' of the parameter. Example: you want to protect the 'program' parameter of IRD I-3. The chain device is named CHN-3. Set the 'enable ID' field for the 'program' parameter to 'CHN-3.locked' and the 'enable Val' to 'false'. The parameter will be grayed out and treated as read-only as soon as the chain device is locked.

The satellite selection (and in some configurations polarization setting too) may be locked even if the RX-ChannelSelect itself is unlocked. This happens if another RX-ChannelSelect device is locked and set to use the same antenna than this one. In this case we must not select another satellite because this would move the antenna and therefore would break the running reception. For this reason, the satellite position parameter is locked by 'satLocked' variable, the polarization setting by 'polLocked'.

To control the locked state of a RX-ChannelSelect device from a task-oriented screen, introduce a Parameter-Button element which sends the special value '%IP%' to the 'togglelock' variable of the RX-ChannelSelect device. The user interface expands the '%IP%' to string 'username@ip-address' which is used to identify username and workplace from where the RX-ChannelSelect device gets locked.

The configuration file format

The RX-ChannelSelect's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-ChannelSelect devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

- *column --- contents --- remark*
- 1 --- switch position / matrix input --- If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
- 2 --- antenna name --- The antenna name must not contain spaces, commas ot pipe characters ('|').

- 3 --- polarization --- A one letter code, one of the follwing**H** for horizontal **V** for vertical **P** for polarization offset With the latter selection, the RX-ChannelSelect device commands the Antenna-Pointing device of the antenna to select the polarization plane by turning the feed.
- 4 --- LNB LO frequency for this port --- The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
- 5 --- **<** or **>** --- If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the nect column. '**<** ' works the other way round.
- 6 --- frequency limit for this port --- The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#
#  Example configuration file for a RX-ChannelSelect device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |               |
#  |   antenna  |  LO freq. [MHz] |
#  |   |       | |             |

   1   ARD     H  9750.0       < 11850.0
   2   ARD     V  9750.0       < 11850.0
   3   ARD     H  10600.0       > 11850.0
   4   ARD     V  10600.0       > 11850.0

   5   NDR     H  9750.0       < 11850.0
   6   NDR     V  9750.0       < 11850.0
   7   NDR     H  10600.0       > 11850.0
   8   NDR     V  10600.0       > 11850.0

   9   EBU     H  10000.0       > 0.0
  10   EBU     V  10000.0       > 0.0

  11   TTC     P  10000.0       > 0.0
  12   TTC     P  10000.0       > 0.0

  13   ASTRA    H  9750.0       < 11850.0
  14   ASTRA    V  9750.0       < 11850.0
  15   ASTRA    H  10600.0       > 11850.0
  16   ASTRA    V  10600.0       > 11850.0

  17   SNGR    H  9750.0       < 11850.0
  18   SNGR    V  9750.0       < 11850.0
  19   SNGR    H  10600.0       > 11850.0
  20   SNGR    V  10600.0       > 11850.0

  21   I605    V  9750.0       > 0.0

  22   W1      V  10000.0       > 0.0
```

The RX-ChannelSelect device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all IRD-CHain devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.configFile	TEXT	SAVE SETUP	StringRange
config.refChannel	TEXT	SAVE SETUP	StringRange
antenna	CHOICE		EnumRange (?)
satellite	CHOICE		EnumRange (?)
channel	CHOICE		EnumRange (?)
polarization	CHOICE		EnumRange (H V)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
fec	CHOICE		EnumRange (?)
comment	TEXT		StringRange
edit.satellite	CHOICE		EnumRange (?)
edit.channel	CHOICE		EnumRange (?)
edit.rxpol	CHOICE		EnumRange (H V)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (H V)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
edit.fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 AUTO)

name	*type*	*flags*	*range*
edit.comment	TEXT		StringRange
edit.newchannel	TEXT		StringRange
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
reset	TEXT		StringRange
alias	TEXT	SAVE	StringRange
quality	TEXT	R/O	StringRange R/O
signal	BOOLEAN	R/O	BooleanRange R/O (true false)
locked	TEXT	R/O	StringRange R/O
satLocked	TEXT	R/O	StringRange R/O
polLocked	TEXT	R/O	StringRange R/O
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IRD summary)

9.561 RX-LineSettings-CDB-DBonly

The RX-LineSettings-CDB logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization. The device complements the TX-LineSettings-CDB logical device which provides a similar function for transmit chains.

This is a special version of the device, which provides only access to the database.

Heart of the RX-LineSettings-CDB device is a central database (CDB) storing the line settings for all satellite channels used in the system. In a NMS environment the database is located in the NMS server, providing an synchronization mechanism which creates local (read-only) copies of the database for offline operation of the VLCs. In a NMS environment, the sat-nms client software looks up the channel lists directly at the NMS database server which significantly reduces the amount of data which has to be exchanged between NMS and VLC. In a M&C environment the CDB is located at the M&C server, no synchronization is required in this case.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The line settings database editor page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (database edit)

With this page of the device window of the RX-LineSettings-CDB device channel definitions are edited, copied or deleted. All RX-LineSettings devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the RX-LineSettings-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete
  2. Click 'DELETE'

Local database copy

The following paragraph only applies to NMS installations. M&C installation manage their CDB locally at the M&C server and do not require a local copy of the data.

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the RX-LineSettings-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a RX-LineSettings-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC upadate it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the RX-LineSettings-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the RX-LineSettings-CDB device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

Just leave the configuration parameters empty.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
config.filterBySatellite	CHOICE	SAVE SETUP	EnumRange (NO YES)
antenna	CHOICE		EnumRange (unknown)
satellite	CHOICE		EnumRange (unknown)
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	TEXT	R/O	StringRange R/O
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
channel	CHOICE		EnumRange (@@@channels.m)
reset	TEXT		StringRange

name	*type*	*flags*	*range*
locked	BOOLEAN	SAVE	BooleanRange (true false)
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y L R)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y L R)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.bitRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.useSymbolRate	CHOICE		EnumRange (SYMBOLRATE BITRATE)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
edit.packetSize	CHOICE		EnumRange (188 204)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.frameSize	CHOICE		EnumRange (SHORT NORMAL)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.videoRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.videoRateAuto	CHOICE		EnumRange (FIXED AUTO)

name	*type*	*flags*	*range*
edit.profile	CHOICE		EnumRange (MPEG2-MP@ML-420 MPEG2-MP@LL-420 MPEG2-HP@ML-422 MPEG2-HP@LL-422 MPEG2-SP@LL-420 MPEG2-SP@ML-420 MPEG2-MP@H14-420 MPEG2-MP@HL-420 MPEG2-HP@H14-422 MPEG2-HP@HL-422 MPEG2-HP@HL-420 MPEG2-HP@H14-420 MPEG2-HP@ML-420 MPEG4-BP@L2-420 MPEG4-BP@L3-420 MPEG4-BP@L4-420 MPEG4-MP@L3-420 MPEG4-MP@L4-420 MPEG4-MP@L4.2-420 MPEG4-HP@L3-420 MPEG4-HP@L4-420 MPEG4-HP@L4.2-420 MPEG4-HP@L3-422 MPEG4-HP@L4-422 MPEG4-HP@L4.2-422)
edit.eirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.reducedEirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.autoLineUp	CHOICE		EnumRange DIS (OFF ON)
edit.audioRate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio2Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio3Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio4Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audioChannels	CHOICE		EnumRange (1 2 3 4)
edit.comment	TEXT		StringRange
edit.user	TEXT		StringRange
edit.brEnable	TEXT	R/O	StringRange R/O
edit.srEnable	TEXT	R/O	StringRange R/O
edit.vrEnable	TEXT	R/O	StringRange R/O
edit.ar2Enable	TEXT	R/O	StringRange R/O
edit.ar3Enable	TEXT	R/O	StringRange R/O
edit.ar4Enable	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

9.562 RX-LineSettings-CDB

The RX-LineSettings-CDB logical device controls the basic parameters of an IRD and an input switch for the IRD which selects frequency band and polarization. The device complements the TX-LineSettings-CDB logical device which provides a similar function for transmit chains.

RX-LineSettings-CDB supersedes the RX-ChannelSelect-CDB device, which no longer should be used for new applications. The RX-LineSettings-CDB device supports a larger number of parameters, may be used in M&C and in NMS environments and it supports an alternative operation mode / work flow. The name RX-LineSettings-CDB reflects that the set of parameters managed by the device extend the standard channel parameters.

Heart of the RX-LineSettings-CDB device is a central database (CDB) storing the line settings for all satellite channels used in the system. In a NMS environment the database is located in the NMS server, providing an synchronization mechanism which creates local (read-only) copies of the database for offline operation of the VLCs. In a NMS environment, the sat-nms client software looks up the channel lists directly at the NMS database server which significantly reduces the amount of data which has to be exchanged between NMS and VLC. In a M&C environment the CDB is located at the M&C server, no synchronization is required in this case.

Although all individual IRD parameters are directly accessible with the RX-LineSettings-CDB device, the typical way to operate the device is to select a satellite channel from the list and to select the proper input signal for the IRD. The RX-LineSettings-CDB device assists the operator in this task with the following functions:

- The list of available input signals is dynamically generated by the device, the list entries are composed of "antenna name", "frequency band", "polarization" and satellite name. E.g. **ANTENNA3-11Y-ASTRA19.2** denotes an 11GHz vertical signals, received from the ASTRA19.2 satellite through ANTENNA3. With motorized antennas, the satellite name is derived from the antenna's Antenna-Pointing device, telling the actually commanded

satellite name.
- If you select a satellite channel from the database, parameters like frequency, symbol rate, DVB-S/DVB-S2 selection etc. are taken from the database and commanded to the IRD. Additionally the list of available inputs is confined to those signals which match the frequency band and polarization plane of the selected channel. You may remove this confinement at any time by selecting the "manual" satellite channel.
- The RX-LineSettings-CDB device accounts for the LO-frequency defined for a selected input and sets the proper receive frequency at the IRD (in terms of L-band frequency). If you change the input, the LO-frequency may change as well and the RX-LineSettings-CDB device commands a new frequency setting to the IRD.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device.
- --- The line settings database editor page.
- --- The fault page.
  - **IRD Lock** : The associated IRD is out of lock
  - **Config File** : The config file is not found, not readable or in a wrong format.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

The TX-LineSettings-CDB device supports two different operation modes / work flow settings which are selected by the 'filterBySatellite' configuration parameter.

With 'filterBySatellite=NO' (the default mode), selecting an already defined satellite channel is a simple 2-step procedure:

1. Select the desired satellite channel from the list.
2. Select the input signal.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The RX-LineSettings-CDB device forwards these settings to the IRD. The input selection list gets confined to the inputs matching the frequency band of the entered frequency.

With 'filterBySatellite=YES', the 'channel' field only shows the channels of the satellite the selected antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the TX-LineSettings-CDB's device screen may me used or the antenna may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. The typical sequence of operation in this mode is as follows:

1. Select the antenna to receive from (field "antenna")
2. Select the satellite to receive with this antenna if this needs to be changed (field "satellite")
3. Select the satellite channel to receive. (field "channel", shows only channels from the selected satellite) Selecting the satellite channel will automatically select the appropriate l-band matrix input and set the IRD parameters. Only if the input selection is ambiguous, e.g. if the satellite can be received through a fixed and a motor driven antenna at the same time, the input selection must be done explicitly.

The 'filterBySatellite=YES' operation mode is only possible in M&C environments.

Operation (database edit)

With the second page of the device window of the RX-LineSettings-CDB device channel definitions are edited, copied or deleted. All RX-LineSettings devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the RX-LineSettings-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
    1. Select the channel to modify
    2. Edit the settings
    3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
    1. Click 'READ' to fill the fields from actual receive settings.
    2. Optionally complete the transmit parameters for the channel.
    3. Enter the desired satellite / channel name.
    4. Click 'SAVE'
- Delete a channel:
    1. Select the channel to delete
    2. Click 'DELETE'

Local database copy

The following paragraph only applies to NMS installations. M&C installation manage their CDB locally at the M&C server and do not require a local copy of the data.

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the RX-LineSettings-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a RX-LineSettings-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the RX-LineSettings-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the RX-LineSettings-CDB device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-LineSettings-CDB device with the IRD and the input switch.

parameter	***description***

parameter	*description*
switchId	Enter the message identifier here of the parameter which shall receive the switch setting, e.g. "MATRIX.12.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
configFile	The name of the configuration file, the RX-LineSettings-CDB device reads the matrix input assignments from. You may use a common configuration file for all RX-LineSettings-CDB devices in the system or an individual file. The file format is described at the end of this chapter.
frequencyId	The parameter with the message ID entered receives the receive frequency setting from the device.
fecId	The message ID for setting the FEC
symbolRateId	The message ID for setting the symbol rate
dvbModeId	The message ID for setting the DVB-S/DVB-S2 mode. For most IRDs use the 'dvbMode' parameter for this. For the Scopus IRD2900 please read it's driver documentation for this parameter.
irdLockFaultId	The message ID for the lock fault flag of the IRD.
filterBySatellite	Selects the workflow / operation mode of the device. "NO" selects the standard, unfiltered mode: The 'channel' parameter always offers all line settings records stored in the database. "YES" switch to filtered mode, the 'channel' parameter shows only the channel matching the actual satellite addressed by 'satellite'.

To make the RX-LineSettings-CDB device work, at least the frequencyId and symbolRateId fields must be set to the message IDs of the corresponding IRD parameters. If some other parameters are not supported by the IRD, leave the message ID fields empty, the same applies to the lock fault if a lock fault signalling in the RX-LineSettings-CDB device is not desired.

The configuration file format

The RX-LineSettings's configuration file is a plain text file which defines the matrix / switch input assignments. As this is a system wide assignment, you usually will edit one common configuration file for all RX-LineSettings-CDB devices connected to one input matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	*contents*	*remark*

column	*contents*	*remark*
1	switch position / matrix input	If the input numbering starts at one or at zero, depends on the matrix / switch device driver.
2	antenna name	The antenna name must not contain spaces, commas ot pipe characters ('
3	polarization	A one letter code, one of the following: **X** for horizontal **Y** for vertical
4	LNB LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	**<** or **>**	If this column contains '**<** ', the switch position is used for frequencies **below** the limit contained in the next column. '**<** ' works the other way round.
6	frequency limit for this port	The frequency limit must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#  Example configuration file for a RX-LineSettings-CDB device
#
#
#  input no.    polarization      freq. limit [MHz]
#  |          |              |
#  |  antenna  |  LO freq. [MHz]  |
#  |  |       | |            |

 01  ARD     X  9750.0      < 11800.0
 02  ARD     Y  9750.0      < 11800.0
 03  ARD     X  10600.0      > 11800.0
 04  ARD     Y  10600.0      > 11800.0

 05  NDR     X  9750.0      < 11800.0
 07  NDR     X  10600.0      > 11800.0

 17  SNGR    X  9750.0       < 11800.0
 18  SNGR    Y  9750.0       < 11800.0
 19  SNGR    X  10600.0      > 11800.0
 20  SNGR    Y  10600.0      > 11800.0

 09  EBU     X  10000.0      < 11800.0
 10  EBU     Y  10000.0      < 11800.0

 11  W1      X  9750.0      < 11800.0
 12  W1      Y  9750.0      < 11800.0
 23  W1      X  10600.0      > 11800.0
 24  W1      Y  10600.0      > 11800.0

 13  ASTRA    X  9750.0       < 11800.0
 14  ASTRA    Y  9750.0       < 11800.0
 15  ASTRA    X  10600.0      > 11800.0
 16  ASTRA    Y  10600.0      > 11800.0

 21  EB_12V   Y  10600.0      > 11800.0

 22  W1_V     Y  10000.0      < 11800.0
```

The RX-LineSettings-CDB device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.irdLockFaultId	TEXT	SAVE SETUP	StringRange
config.filterBySatellite	CHOICE	SAVE SETUP	EnumRange (NO YES)
antenna	CHOICE		EnumRange (unknown)
satellite	CHOICE		EnumRange (unknown)
input	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
polarization	TEXT	R/O	StringRange R/O
comment	TEXT	SAVE	StringRange
rxFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
channel	CHOICE		EnumRange (@@@channels.m)

name	*type*	*flags*	*range*
reset	TEXT		StringRange
locked	BOOLEAN	SAVE	BooleanRange (true false)
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y L R)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y L R)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.bitRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.useSymbolRate	CHOICE		EnumRange (SYMBOLRATE BITRATE)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
edit.packetSize	CHOICE		EnumRange (188 204)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.frameSize	CHOICE		EnumRange (SHORT NORMAL)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.videoRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.videoRateAuto	CHOICE		EnumRange (FIXED AUTO)

name	*type*	*flags*	*range*
edit.profile	CHOICE		EnumRange (MPEG2-MP@ML-420 MPEG2-MP@LL-420 MPEG2-HP@ML-422 MPEG2-HP@LL-422 MPEG2-SP@LL-420 MPEG2-SP@ML-420 MPEG2-MP@H14-420 MPEG2-MP@HL-420 MPEG2-HP@H14-422 MPEG2-HP@HL-422 MPEG2-HP@HL-420 MPEG2-HP@H14-420 MPEG2-HP@ML-420 MPEG4-BP@L2-420 MPEG4-BP@L3-420 MPEG4-BP@L4-420 MPEG4-MP@L3-420 MPEG4-MP@L4-420 MPEG4-MP@L4.2-420 MPEG4-HP@L3-420 MPEG4-HP@L4-420 MPEG4-HP@L4.2-420 MPEG4-HP@L3-422 MPEG4-HP@L4-422 MPEG4-HP@L4.2-422)
edit.eirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.reducedEirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.autoLineUp	CHOICE		EnumRange DIS (OFF ON)
edit.audioRate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio2Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio3Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio4Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audioChannels	CHOICE		EnumRange (1 2 3 4)
edit.comment	TEXT		StringRange
edit.user	TEXT		StringRange
edit.brEnable	TEXT	R/O	StringRange R/O
edit.srEnable	TEXT	R/O	StringRange R/O
edit.vrEnable	TEXT	R/O	StringRange R/O
edit.ar2Enable	TEXT	R/O	StringRange R/O
edit.ar3Enable	TEXT	R/O	StringRange R/O
edit.ar4Enable	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRD lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

9.563 RadeusLab-ACU-8200

This page describes the device driver and the device window for the RadeusLab-ACU-8200 via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the tracking state.
- --- Contains the target list
- --- Contains the tracking parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- jogDuration --- Sets the duration in ms for a jog step. Default value is 500 ms.
- azLoLimit --- Sets the lower azimuth travel limit.
- azHiLimit --- Sets the upper azimuth travel limit.
- elLoLimit --- Sets the lower elevation travel limit.
- elHiLimit --- Sets the upper elevation travel limit.
- plLoLimit ---Sets the lower polarization travel limit.
- plHiLimit --- Sets the upper polarization travel limit.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.dbVersion	TEXT	R/O	StringRange R/O
info.antennaName	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.poslog	TEXT	R/O	StringRange R/O
internal.flt1	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt2	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt3	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt4	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt5	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt6	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt7	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt8	HEX	R/O	HexRange R/O (0 .. 0)
internal.flt9	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck1	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck2	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck3	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck4	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck5	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck6	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck7	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck8	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltAck9	HEX	R/O	HexRange R/O (0 .. 0)
targetId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state	TEXT	R/O	StringRange R/O
internal.zero	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
initialized	TEXT	R/O	StringRange R/O
config.jogduration	INTEGER	SAVE SETUP	IntegerRange (0 .. 3600000)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
manual	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.movingAz	BOOLEAN	R/O	BooleanRange R/O (true false)
state.movingEl	BOOLEAN	R/O	BooleanRange R/O (true false)
state.movingPl	BOOLEAN	R/O	BooleanRange R/O (true false)
state.onTarget	BOOLEAN	R/O	BooleanRange R/O (true false)
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (1 .. 100)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.rfInput	TEXT	R/O	StringRange R/O
state.pol1	TEXT	R/O	StringRange R/O
state.pol2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
trackMode	CHOICE		EnumRange (UNUSED LOOK ANGLES LONGITUDE PREDICTIVE STEPTRACK TLE IESS TLE-STEP IESS-STEP TLE-GROUP)
interval	INTEGER		IntegerRange (1 .. 1638)
stepSize	INTEGER		IntegerRange (1 .. 20)
frequency	FLOAT		DoubleRange (950.000 .. 40000.000)
deadband	INTEGER		IntegerRange (1 .. 20)
maxStepSize	INTEGER		IntegerRange (10 .. 50)
maxRetries	INTEGER		IntegerRange (1 .. 10)
orbit	FLOAT		DoubleRange (0.000 .. 0.000)
beamWidth	FLOAT		DoubleRange (0.000 .. 0.000)
levelThreshold	FLOAT		DoubleRange (-20.0 .. 0.0)
levelOffset	FLOAT		DoubleRange (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (2 .. 100)
target.save	INTEGER	NOPRESET	IntegerRange (2 .. 100)
target.delete	INTEGER	NOPRESET	IntegerRange (2 .. 100)
goto	INTEGER	NOPRESET	IntegerRange (2 .. 100)
grmc	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ Runaway)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ Immobile)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AZ Reversed)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ EncoderFault)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ CCW Soft Limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ CW Soft Limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ Motor Controller)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ Limit Switch Trip)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL Runaway)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL Immobile)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EL Reversed)
faults.12	ALARM	R/O	AlarmFlagRange R/O (EL Encoder Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (EL Lower Soft Limit)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL Upper Soft Limit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (EL Motor Controller)
faults.16	ALARM	R/O	AlarmFlagRange R/O (EL Limit Switch Trip)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PL Runaway)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PL Immobile)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PL Reversed)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PL Encoder Fault)
faults.21	ALARM	R/O	AlarmFlagRange R/O (PL CCW Soft Limit)
faults.22	ALARM	R/O	AlarmFlagRange R/O (PL CW Soft Limit)
faults.23	ALARM	R/O	AlarmFlagRange R/O (PL Limit Switch Trip)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Drive Disabled Ctrl Panel)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Override Drive Cabinet)

name	*type*	*flags*	*range*
faults.27	ALARM	R/O	AlarmFlagRange R/O (Soft Limit Disabled)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Limit Switch Trip)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Power Drive Cabinet)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Signal Level)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Unstable Signal Level)
faults.32	ALARM	R/O	AlarmFlagRange R/O (IOC Communication)
faults.33	ALARM	R/O	AlarmFlagRange R/O (ENC Communication)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RDC Communication)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Jog Panel Communication)
faults.36	ALARM	R/O	AlarmFlagRange R/O (IOC FW Update)
faults.37	ALARM	R/O	AlarmFlagRange R/O (ENC FW Update)
faults.38	ALARM	R/O	AlarmFlagRange R/O (RDC FW Update)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Jog Panel FW Update)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Access Settings)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Access Tracking Hist)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Rcv Communication)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Rcv Not Running)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Rcv Locked)
faults.45	ALARM	R/O	AlarmFlagRange R/O (USB IO 1)

name	*type*	*flags*	*range*
faults.46	ALARM	R/O	AlarmFlagRange R/O (USB IO 2)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Frequency Range)
faults.48	ALARM	R/O	AlarmFlagRange R/O (ICO 2 Communication)
faults.49	ALARM	R/O	AlarmFlagRange R/O (ICO 2 FW Update)
faults.50	ALARM	R/O	AlarmFlagRange R/O (PL Wrong Pos)
faults.51	ALARM	R/O	AlarmFlagRange R/O (TLE Epoch)
faults.52	ALARM	R/O	AlarmFlagRange R/O (IESS Epoch)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Standby Fault)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Drive Cab Network)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Motor Ctrl Reset)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Orbit Limit)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Sun Outage)
faults.58	ALARM	R/O	AlarmFlagRange R/O (AZ Communication)
faults.59	ALARM	R/O	AlarmFlagRange R/O (AZ Not Zeroed)
faults.60	ALARM	R/O	AlarmFlagRange R/O (AZ Download)
faults.61	ALARM	R/O	AlarmFlagRange R/O (EL Communication)
faults.62	ALARM	R/O	AlarmFlagRange R/O (EL Not Zeroed)
faults.63	ALARM	R/O	AlarmFlagRange R/O (EL Download)
faults.64	ALARM	R/O	AlarmFlagRange R/O (PL Communication)
faults.65	ALARM	R/O	AlarmFlagRange R/O (PL Not Zeroed)
faults.66	ALARM	R/O	AlarmFlagRange R/O (PL Download)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Door Open)

name	*type*	*flags*	*range*
faults.68	ALARM	R/O	AlarmFlagRange R/O (PMCU Halted)
faults.69	ALARM	R/O	AlarmFlagRange R/O (PMCU FW Update)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Az Stow Pin)
faults.71	ALARM	R/O	AlarmFlagRange R/O (EL Stow Pin)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Handcrank)

9.564 Radyne-DM240

This page describes the device driver and the device window for the Radyne DM240 Digital Modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DM240.

parameter	*description*
address	The modulator's packet address. Valid addresses are 1..254, several addresses however are reserved by Radyne for special purposes. See the DM240 manual for details.
Serial	Remember to set the line parameters for the serial interface connected to the DM240. The value must correspond to the line speed set at the DM240 frontpanel. The factory setting for the DM240 is *19200,N,8,1* .

Communication protocol / Interface type

The DM240 device driver requires the *Radyne-Comstream-Protocol* . The DM240 uses a RS485 interface for M&C.

Remarks

- The DM240 does not distinguish a local and a remote operation mode.
- The DM240 device driver has been reported to generate communication faults while the M&C computer works at a high load (e.g. backup). Set the "retries" count to a value of 2 or 3 if you encounter this problem.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.srevision	TEXT	R/O	StringRange R/O
info.alarmMask	HEX	R/O	HexRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
modemType	CHOICE		EnumRange (DVB DIRECPC DIRECTV)
refClkSrc	CHOICE		EnumRange (INT EXT)
refClkFreq	FLOAT		DoubleRange (0.256000 .. 10.000000)
tx.mod.dataRate	INTEGER		IntegerRange (1000000 .. 45000000)
tx.mod.symbolRate	INTEGER		IntegerRange (1000000 .. 52000000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK 8PSK 16QAM)

name	*type*	*flags*	*range*
tx.mod.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9)
tx.mod.rollOff	CHOICE		EnumRange (0.35 0.25 0.20)
tx.mod.clockSource	CHOICE		EnumRange (SCTE SCT)
tx.mod.cwMode	CHOICE		EnumRange (MOD CW DUAL OFFS FIR+ FIR-)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 204 NONE DIRECPC DIRECTV)
tx.ifc.clockPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.hardware	CHOICE		EnumRange (SERIAL PARALLEL ASI-NORM ASI-NULL G.703-E3 G.703-T3 STS-1 HSSI PAR DVB PAR M2P NONE DIRECTV OC3 STM-1 G.703E2 G.703T2BAL G.703T2UNBL G.703E1BAL G.703E1UNBL G.703T1AMI G.703T1B8ZS)
tx.frequency	FLOAT		DoubleRange (50.0000 .. 180.0000)
tx.power	FLOAT		DoubleRange (-20.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX oversample PLL)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FPGA config)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IF synth PLL)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ext. reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Terrestrial clock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Terrestrial data)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (R-S FIFO)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Terrestrial framing)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Supply -12V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Supply +12V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Supply +5V)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.on	CHOICE	R/O	EnumRange R/O (OFF ON)

9.565 Radyne-DMD2050

This page describes the device driver and the device window for the Radyne DMD 2050 Satellite Modem.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The Automatic Uplink Power Control settings.
- --- The data interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DMD2050.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field

is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- tx.loFreq --- Sets the TX LO frequency (MHz) used to convert the transmitted signal from L-band to RF. Be sure to set this to 0 if the modem is used with 70MHz IF.
- tx.sideBand --- Selects the side band used with the L-band to RF conversion. Select "HIGH-SIDE" if the carrier frequency is above the LO, "LOW-SIDE" if below.
- tx.lband.10MHz --- Switches a 10MHz reference signal on or off at the L-band output.
- tx.lband.voltageEnable --- Switches the BUC supply voltage on or off at the L-band output.
- rx.loFreq --- Sets the RX LO frequency (MHz) used to convert the received signal to L-band. Be sure to set this to 0 if the modem is used with 70MHz IF.
- rx.sideBand --- Selects the side band used with the RF to L-band down conversion. Select "HIGH-SIDE" if the carrier frequency is above the LO, "LOW-SIDE" if below.
- rx.lband.10MHz --- Switches a 10MHz reference signal on or off at the L-band input.
- rx.lband.voltageEnable --- Switches the LNB supply voltage on or off at the L-band input.
- rx.lband.voltageSelect --- Selects the supply voltage to be applied at the L-band input. (13/15/18/20V)

Remarks

- The LO frequency calculation (L-band to RF an vice versa) is done in the sat-nms device driver, not in the device itself. As a result, the modem itself will show L-band frequency at the front panel, not the RF frequency displayed in the sat-nms software. The LO frequency and sideband selection configuration parameters are stored on the VLC, not in the device.
- The tx.modemType and rx.modemType parameters in fact are sort of macros, setting the modem to default values suitable for the selected operation mode. The sat-nms driver does not store these parameters in preset to avoid any interference between the initializations done by these commands and the parameters set by the preset.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.psu5V	FLOAT	R/O	DoubleRange R/O (0.0 .. 55.0)
info.psu12V	FLOAT	R/O	DoubleRange R/O (0.0 .. 55.0)
info.psum12V	FLOAT	R/O	DoubleRange R/O (-55.0 .. 0.0)
info.revision	TEXT	R/O	StringRange R/O
info.lnbVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 55.0)
info.lnbCurrent	FLOAT	R/O	DoubleRange R/O (0.0 .. 800.0)
info.bucVoltage	FLOAT	R/O	DoubleRange R/O (0 .. 55)
info.bucCurrent	FLOAT	R/O	DoubleRange R/O (0 .. 800)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.tx.loFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.tx.sideBand	CHOICE	SAVE SETUP	EnumRange (HIGH-SIDE LOW-SIDE)
config.tx.lband.10MHz	CHOICE	SETUP	EnumRange (DISABLE ENABLE)
config.tx.lband.voltageEnable	CHOICE	SETUP	EnumRange (DISABLE ENABLE)
config.rx.loFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.rx.sideBand	CHOICE	SAVE SETUP	EnumRange (HIGH-SIDE LOW-SIDE)

name	*type*	*flags*	*range*
config.rx.lband.10MHz	CHOICE	SETUP	EnumRange (DISABLE ENABLE)
config.rx.lband.voltageEnable	CHOICE	SETUP	EnumRange (DISABLE ENABLE)
config.rx.lband.voltageSelect	CHOICE	SETUP	EnumRange (13V 15V 18V 20V)
tx.modemType	CHOICE	NOPRESET	EnumRange (CLOSED-NET IDR IBS DROP-INSERT DVB-SAT MIL-STD188PT165-A RFM EBEM)
tx.on	CHOICE	NOPRESET	EnumRange (OFF ON AUTO VSAT RTS)
tx.power	FLOAT		DoubleRange (-45.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.mod.dataRate	FLOAT		DoubleRange (0.004800 .. 52.000000)
tx.mod.symbolRate	FLOAT	R/O	DoubleRange R/O (0.009600 .. 10.000000)
tx.mod.fec	CHOICE		EnumRange (NONE VITERBI-1/2 VITERBI-2/3 VITERBI-3/4 VITERBI-5/6 VITERBI-7/8 SEQUENTIAL-1/2 SEQUENTIAL-3/4 SEQUENTIAL-7/8 TRELLIS-2/3 TRELLIS-3/4 TRELLIS-5/6 TRELLIS-7/8 TRELLIS-8/9 COMSTREAM-3/4 TPC-793/2D TPC-495/3D TPC-1/2 TPC-3/4 TPC-7/8 TPC-21/44 TPC-750 TPC-875 TPC-288 TPC-7/8-S TPC-5/16 LDPC-1/2 LDPC-2/3 LDPC-3/4 LDPC-4/5 LDPC-5/6 LDPC-8/9 LDPC-9/10 LDPC-3/5 FLEX-1/2 FLEX-2/3 FLEX-3/4 FLEX-7/8 FLEX-19/20)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK PSK8 QAM16 OQPSK RFM QAM8 APSK16)

name	*type*	*flags*	*range*
tx.mod.testPattern	CHOICE		EnumRange (NORMAL TEST2047 TEST-PATTERN2-TO15-MINUS1 TEST-PATTERN2-TO23-MINUS1)
tx.frm.type	CHOICE		EnumRange (NONE 96K-IDR IBS EF-AUPC DVB EDMAC SCC 96K EFFICIENT-DNI EBEM)
tx.rs.bypass	CHOICE		EnumRange (ENCODE BYPASS)
tx.rs.encoding	CHOICE		EnumRange (RS-126/112 RS-194/178 RS-219/201 RS-225/205 RS-204/188 RS-CUSTOM)
tx.rs.interleave	CHOICE		EnumRange (4 8 12)
tx.ifc.dropMode	CHOICE		EnumRange (DISABLE T1D4 T1ESF PCM-30 PCM-30C PCM-31 PCM-31C SLC96 T1D4S T1ESFS)
tx.ifc.dropMap	TEXT		StringRange
tx.ifc.clockSource	CHOICE		EnumRange (EXT-SCTE INT-SCT)
tx.ifc.clockPhase	CHOICE		EnumRange (NORMAL INVERTED AUTO)
tx.mod.clockSource	CHOICE		EnumRange (INTERNAL SCR)
tx.mod.clockPhase	CHOICE		EnumRange (NORMAL INVERTED)
tx.ifc.dataPhase	CHOICE		EnumRange (NONE TERRESTRIAL BASEBAND TERRESTRIAL+BASEBAND)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORMAL INVERTED)
tx.mod.scramblingEnable	CHOICE		EnumRange (DISABLE ENABLE)

name	*type*	*flags*	*range*
tx.mod.scrambling	CHOICE		EnumRange (NONE IBS-SCRAMBLER V35-IESS V35-CCITT V35-EF-DATA V35-F-C OM73 RS-SCRAMBLER V35-EF-RS TPC-SCRAMBLER DVB-SCRAMBLER EDMAC TPC-IBS TPC-EDMAC V35-COMSTREAM EBEM)
tx.mod.differential	CHOICE		EnumRange (DISABLE ENABLE)
tx.ifc.overheadType	CHOICE		EnumRange (VOICE DATA)
tx.ifc.esc1Gain	FLOAT		DoubleRange (-20 .. 10)
tx.ifc.esc2Gain	FLOAT		DoubleRange (-20 .. 10)
tx.ifc.hardware	CHOICE		EnumRange (G703-B-T1-AMI G703-B-T1-B8ZS G703-B-E1 G703-B-T2 G703-U-E1 G703-U-T2 G703-U-E2 RS422 V35 RS232 HSSI ASI AASI M2P DVB ETHERNET-BRIDGE MIL-STD188PT114-A RS423 G703-U-T3 G703-U-E3 STS1-UNBAL)
tx.aupc.localMode	CHOICE		EnumRange (DISABLED EF-AUPC RADYNE NEAR-SIDE)
tx.aupc.remoteMode	CHOICE		EnumRange (DISABLED EF-AUPC)
tx.aupc.localOnCarrLoss	CHOICE		EnumRange (HOLD NOMINAL MAXIMUM)
tx.aupc.remoteOnCarrLoss	CHOICE		EnumRange (HOLD NOMINAL MAXIMUM)
tx.aupcTrackingRate	CHOICE		EnumRange (0.5dB/min 1.0dB/min 1.5dB/min 2.0dB/min 2.5dB/min 3.0dB/min 3.5dB/min 4.0dB/min 4.5dB/min 5.0dB/min 5.5dB/min 6.0dB/min)
tx.aupc.remoteLoopBack	CHOICE		EnumRange (DISABLE ENABLE)

name	*type*	*flags*	*range*
tx.aupc.remoteTest	CHOICE		EnumRange (DISABLE ENABLE)
tx.aupc.targetEbno	FLOAT		DoubleRange (4.00 .. 20.00)
tx.aupc.minCarrierLevel	FLOAT		DoubleRange (-25.00 .. 0.00)
tx.aupc.maxCarrierLevel	FLOAT		DoubleRange (-25.00 .. 0.00)
tx.aupc.nomCarrierLevel	FLOAT		DoubleRange (-25.00 .. 0.00)
tx.ifc.framingMode	CHOICE		EnumRange (NONE DVB-188 DVB-204)
tx.mod.rollOff	CHOICE		EnumRange (INTELSAT-035 DVB-SAT-025 DVB-SAT-035 MIL-STD188PT165-A DVB-SAT-020)
rx.modemType	CHOICE	NOPRESET	EnumRange (CLOSED-NET IDR IBS DROP-INSERT DVB-SAT MIL-STD188PT165-A RFM EBEM)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.dem.dataRate	FLOAT		DoubleRange (0.004800 .. 52.000000)
rx.dem.symbolRate	FLOAT	R/O	DoubleRange R/O (0.009600 .. 10.000000)
rx.dem.fec	CHOICE		EnumRange (NONE VITERBI-1/2 VITERBI-2/3 VITERBI-3/4 VITERBI-5/6 VITERBI-7/8 SEQUENTIAL-1/2 SEQUENTIAL-3/4 SEQUENTIAL-7/8 TRELLIS-2/3 TRELLIS-3/4 TRELLIS-5/6 TRELLIS-7/8 TRELLIS-8/9 COMSTREAM-3/4 TPC-793/2D TPC-495/3D TPC-1/2 TPC-3/4 TPC-7/8 TPC-21/44 TPC-750 TPC-875 LDPC-1/2 LDPC-2/3 LDPC-3/4 LDPC-4/5 LDPC-5/6 LDPC-8/9 LDPC-9/10 LDPC-3/5 FLEX-1/2 FLEX-2/3 FLEX-3/4 FLEX-7/8 FLEX-19/20)

name	*type*	*flags*	*range*
rx.dem.type	CHOICE		EnumRange (QPSK BPSK PSK8 QAM16 OQPSK RFM QAM8 APSK16)
rx.frm.type	CHOICE		EnumRange (NONE 96K-IDR IBS EF-AUPC DVB EDMAC SCC 96K EFFICIENT-DNI EBEM)
rx.rs.bypass	CHOICE		EnumRange (DECODE BYPASS)
rx.rs.encoding	CHOICE		EnumRange (RS-126/112 RS-194/178 RS-219/201 RS-225/205 RS-204/188 RS-CUSTOM)
rx.rs.interleave	CHOICE		EnumRange (4 8 12)
rx.ifc.insertMode	CHOICE		EnumRange (DISABLE T1D4 T1ESF PCM-30 PCM-30C PCM-31 PCM-31C SLC96 T1D4S T1ESFS)
rx.ifc.insertMap	TEXT		StringRange
rx.dem.clockSource	CHOICE		EnumRange (EXT-SCTE INT-SCT EXT-BNC RX-SAT EXT-IDI)
rx.dem.clockPhase	CHOICE		EnumRange (NORMAL INVERTED)
rx.buf.size	INTEGER		IntegerRange (0 .. 64)
rx.dem.dataInvert	CHOICE		EnumRange (NONE TERRESTRIAL BASEBAND TERRESTRIAL+BASEBAND)
rx.dem.spectrumInvert	CHOICE		EnumRange (NORMAL INVERTED AUTO)
rx.dem.scramblingEnable	CHOICE		EnumRange (DISABLE ENABLE)

name	*type*	*flags*	*range*
rx.dem.scrambling	CHOICE		EnumRange (NONE IBS-DESCRAMBLER V35-IESS V35-CCITT V35-EF-DATA V36-F-C OM73 RS-DESCRAMBLER V35-EF-RS TPC-DESCRAMBLER DVB-DESCRAMBLER EDMAC TPC-IBS TPC-EDMAC V35-COMSTREAM EBEM)
rx.dem.differential	CHOICE		EnumRange (DISABLE ENABLE)
rx.dem.sweepRange	FLOAT		DoubleRange (0 .. 255)
rx.dem.levelLimit	FLOAT		DoubleRange (30 .. 90)
rx.ifc.overheadType	CHOICE		EnumRange (VOICE DATA)
rx.ifc.esc1Gain	FLOAT		DoubleRange (-20 .. 10)
rx.ifc.esc2Gain	FLOAT		DoubleRange (-20 .. 10)
rx.ifc.hardware	CHOICE		EnumRange (G703-B-T1-AMI G703-B-T1-B8ZS G703-B-E1 G703-B-T2 G703-U-E1 G703-U-T2 G703-U-E2 RS422 V35 RS232 HSSI ASI AASI M2P DVB ETHERNET-BRIDGE MIL-STD188PT114-A RS423 G703-U-T3 G703-U-E3 STS1-UNBAL)
rx.ifc.framingMode	CHOICE		EnumRange (NONE DVB-188 DVB-204)
rx.ifc.streaming	CHOICE		EnumRange (PACKET-OUTPUT BYTE-OUTPUT)
rx.dem.sweepDelay	FLOAT		DoubleRange (0.0 .. 6553.5)
rx.dem.ebno.alarmThreshold	FLOAT		DoubleRange (0.00 .. 9.90)
rx.dem.rollOff	CHOICE		EnumRange (INTELSAT-035 DVB-SAT-025 DVB-SAT-035 MIL-STD188PT165-A DVB-SAT-020)
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 25.00)
rx.meter.bufFill	FLOAT	R/O	DoubleRange R/O (0 .. 100)

name	*type*	*flags*	*range*
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (-100 .. 0)
rx.meter.frequencyOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.meter.agc	FLOAT	R/O	DoubleRange R/O (0 .. 990)
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 40.00)
refClkFreq	FLOAT		DoubleRange (0.256 .. 10.000)
refClkSrc	CHOICE		EnumRange (INTERNAL EXTERNAL HIGH-STAB)
tx.mod.cwMode	CHOICE		EnumRange (NORMAL CW DUAL OFFSET POS-FIR NEG-FIR)
loopBack	CHOICE		EnumRange (NONE TERR-TX TERR-RX TERR-TX-RX BASEBAND-TX BASEBAND-RX BASEBAND-TX-RX IF)
internal.rxf	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.txf	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
internal.C1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.C2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.T1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.T2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.R1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.R2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.R3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.R4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.R5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nC1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.nC2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nT1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nT2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nR1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nR2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nR3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nR4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nR5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (-12V Supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (+12V Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (+5V Supply)
faults.04	ALARM	R/O	AlarmFlagRange R/O (+1.5V Rx Supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (+1.5V Tx Supply)
faults.06	ALARM	R/O	AlarmFlagRange R/O (+3.3V Supply)
faults.07	ALARM	R/O	AlarmFlagRange R/O (+20V Supply)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Terr FPGA Cfg)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Codec FPGA Cfg)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Codec Dev Cfg)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Transec Pwr Test)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Tx FPGA Cfg)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Tx DSP Cfg)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Symbol Clk PLL)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tx LB Synth PLL)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Transec Ready)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Tx Terr Clk Act)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Tx Terr Data Act)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Tx Terr AIS)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Clk Src Fallback)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Tx TPC Confl Chk)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Rx FPGA Cfg)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Rx DSP Cfg)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Rx Signal Lock)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Rx Input Level)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Rx Frame Lock)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Rx MFrame Lock)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (Rx LB Synth PLL)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Buf Underflow)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Buf near emtpy)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Buf near full)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Buf Overflow)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Rx Data Act)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Rx Sat AIS)
faults.35	ALARM	R/O	AlarmFlagRange R/O (IFEC Lock)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Rx TPC Confl Chk)
faults.37	ALARM	R/O	AlarmFlagRange R/O (OFEC Lock)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Rx Interleaver)
faults.39	ALARM	R/O	AlarmFlagRange R/O (RS UncorrWord)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Eb/No)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Rx Level)

9.566 Radyne-DVB3030

This page describes the device driver and the device window for the Radyne DVB-3030 Digital Video Broadcast Modulator.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global settings.
- --- The modulator settings, transmit clocking.
- --- The TX interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DVB3030.

parameter	***description***
address	The modulator's packet address. Valid addresses are 1..254, several addresses however are reserved by Radyne for special purposes. See the DVB3030 manual for details.
control	You can select if the modulator shall accept parameter changes from the from panel, the M&C computer or via Ethernet. The device driver will generate a fault unless the DVB3030 is set to "COMPUTER" control.
Serial	Remember to set the line parameters for the serial interface connected to the DVB3030. The proper setting is *9600,N,8,1* regardless of the value set at the unit's front panel (this setting refers to the terminal port, not to the remote control port of the DVB3030).

Communication protocol / Interface type

The DVB3030 device driver requires the *Radyne-Comstream-Protocol* . The DVB3030 uses a RS485 interface for M&C.

Installation

Before the sat-nms software will recognize the modulator, you have to set the communication parameters at the unit's front panel. Set the device address to the chosen value and then place the device in "COMPUTER" control mode.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.control	CHOICE	SETUP	EnumRange (LOCAL TERMINAL COMPUTER TCP/IP)
info.srevision	TEXT	R/O	StringRange R/O
info.alarmMask	HEX	R/O	HexRange R/O (0 .. 0)
info.maskEnable	CHOICE	R/O	EnumRange R/O (ENABLE DISABLE)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
modemType	CHOICE		EnumRange (DVB DSS)
refClkSrc	CHOICE		EnumRange (INT EXT)
refClkFreq	FLOAT		DoubleRange (1.000000 .. 10.000000)
tx.mod.dataRate	INTEGER		IntegerRange (1000000 .. 45000000)
tx.mod.symbolRate	INTEGER		IntegerRange (1000000 .. 52000000)
tx.mod.type	CHOICE		EnumRange (QPSK BPSK)
tx.mod.fec	CHOICE		EnumRange (NONE 1/2 2/3 3/4 5/6 6/7 7/8)
tx.mod.rollOff	CHOICE		EnumRange (0.35 0.20)
tx.mod.clockSource	CHOICE		EnumRange (SCTE SCT)
tx.mod.cwMode	CHOICE		EnumRange (MOD CW DUAL OFFS FIR+ FIR-)
tx.mod.spectrumInvert	CHOICE		EnumRange (NRM INV)
tx.ifc.framingMode	CHOICE		EnumRange (188 204 NONE)
tx.ifc.clockPhase	CHOICE		EnumRange (NRM INV)
tx.ifc.dataPhase	CHOICE		EnumRange (NRM INV)

name	*type*	*flags*	*range*
tx.ifc.hardware	CHOICE		EnumRange (SERIAL PARALLEL ASI-NORM ASI-NULL G.703-E3 G.703-T3 STS-1 HSSI PAR DVB PAR M2P)
tx.frequency	FLOAT		DoubleRange (50.0000 .. 90.0000)
tx.power	FLOAT		DoubleRange (-20.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX output power level)
faults.02	ALARM	R/O	AlarmFlagRange R/O (TX oversample PLL lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IF synth PLL lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ext reference PLL lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Frm sync Xilinx config)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Frm interl Xilinx config)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mod map Xilinx config)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Internal clock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Terrestrial clock)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Terrestrial data)
faults.11	ALARM	R/O	AlarmFlagRange R/O (R-S FIFO empty)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Interleaver frame count)
faults.13	ALARM	R/O	AlarmFlagRange R/O (DVB frame synch.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Supply -12V)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Supply +12V)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Supply +5V)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (RAM/ROM)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Remote access)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.on	CHOICE	R/O	EnumRange R/O (OFF ON)

9.567 Radyne-RCS11

There is no special online help for this type of device. Please refer to the general description of device windows instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.configuration	TEXT	R/O	StringRange R/O
info.firmwareVersion	HEX	R/O	HexRange R/O (0 .. 0)
info.firmwareDate	TEXT	R/O	StringRange R/O
info.firmwareName	TEXT	R/O	StringRange R/O
config.alarmMask	HEX	SETUP	HexRange (0 .. 0)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
automatic	CHOICE		EnumRange (MANUAL AUTO)
forceA	TEXT		StringRange

name	*type*	*flags*	*range*
forceB	TEXT		StringRange
tx.position	CHOICE	R/O	EnumRange R/O (A B ?)
tx.faultDelay	FLOAT		DoubleRange (0.1 .. 300.0)
tx.okDelay	FLOAT		DoubleRange (0.1 .. 300.0)
tx.faultA	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.forcedA	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.faultB	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.forcedB	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.position	CHOICE	R/O	EnumRange R/O (A B ?)
rx.faultDelay	FLOAT		DoubleRange (0.1 .. 300.0)
rx.okDelay	FLOAT		DoubleRange (0.1 .. 300.0)
rx.faultA	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.forcedA	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.faultB	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.forcedB	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Switch PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch firmware)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch NV RAM)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch internal)

9.568 Radyne-SFC1450

This page describes the device driver and the device window for the Radyne Comstream SFC6400 and SFC1450 upconverters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	***description***
address	The device address as set at the device's front panel. The factory preset address of the upconverter is "01".
inputAtten	The input attenuator of the device can be controlled with this parameter.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,N,8,1* .

Remarks

Please note, that the M&C software expects the upconverter to operate the serial interface with "ECHO ON" (which is the factory preset). There will be a communication fault if the the upconverter is set to "ECHO OFF" at the front panel.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.configuration	TEXT	R/O	StringRange R/O
info.reference	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.inputAtten	INTEGER	SETUP	IntegerRange (0 .. 30)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (14000.000 .. 14500.000)
reset	TEXT	NOPRESET	StringRange
internal.maxgain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.mingain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.frqfrac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (General)

9.569 Raritan-EMX

This page describes the device driver and the device window for the Raritan EMX rack management solution including up to 16 sensors

The driver recognizes the number and type of installed sensors and sets their ranges accordingly

The driver was tested with the following model:

- Model: EMX2-111
- SW: 3.5.0.5-45371

The driver was tested with the following sensors:

- Temperature
- Absolute Humidity
- Relative Humidity
- Contact

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all sensor measurements as well as their thresholds
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.sensors	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
info.sens.01.serial	TEXT	R/O	StringRange R/O
info.sens.02.serial	TEXT	R/O	StringRange R/O
info.sens.03.serial	TEXT	R/O	StringRange R/O
info.sens.04.serial	TEXT	R/O	StringRange R/O
info.sens.05.serial	TEXT	R/O	StringRange R/O
info.sens.06.serial	TEXT	R/O	StringRange R/O
info.sens.07.serial	TEXT	R/O	StringRange R/O
info.sens.08.serial	TEXT	R/O	StringRange R/O
info.sens.09.serial	TEXT	R/O	StringRange R/O
info.sens.10.serial	TEXT	R/O	StringRange R/O
info.sens.11.serial	TEXT	R/O	StringRange R/O
info.sens.12.serial	TEXT	R/O	StringRange R/O
info.sens.13.serial	TEXT	R/O	StringRange R/O
info.sens.14.serial	TEXT	R/O	StringRange R/O
info.sens.15.serial	TEXT	R/O	StringRange R/O
info.sens.16.serial	TEXT	R/O	StringRange R/O
sens.01.name	TEXT		StringRange
sens.02.name	TEXT		StringRange
sens.03.name	TEXT		StringRange
sens.04.name	TEXT		StringRange
sens.05.name	TEXT		StringRange
sens.06.name	TEXT		StringRange

name	*type*	*flags*	*range*
sens.07.name	TEXT		StringRange
sens.08.name	TEXT		StringRange
sens.09.name	TEXT		StringRange
sens.10.name	TEXT		StringRange
sens.11.name	TEXT		StringRange
sens.12.name	TEXT		StringRange
sens.13.name	TEXT		StringRange
sens.14.name	TEXT		StringRange
sens.15.name	TEXT		StringRange
sens.16.name	TEXT		StringRange
sens.01.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.02.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.03.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.04.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.05.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.06.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.07.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.08.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.09.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.10.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.11.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.12.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.13.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.14.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.15.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
sens.16.meas	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
sens.01.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
sens.14.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.loWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.hiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
sens.11.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.loCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.hiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
sens.08.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.defLoWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.defHiWarnThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.02.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.03.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.04.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
sens.05.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.05.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.06.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.07.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.08.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.09.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.10.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.11.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.12.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.13.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.14.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.15.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.defLoCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.16.defHiCritThreshold	FLOAT		DoubleRange (0.00 .. 0.00)
sens.01.useDefault	CHOICE		EnumRange (ON OFF)
sens.02.useDefault	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
sens.03.useDefault	CHOICE		EnumRange (ON OFF)
sens.04.useDefault	CHOICE		EnumRange (ON OFF)
sens.05.useDefault	CHOICE		EnumRange (ON OFF)
sens.06.useDefault	CHOICE		EnumRange (ON OFF)
sens.07.useDefault	CHOICE		EnumRange (ON OFF)
sens.08.useDefault	CHOICE		EnumRange (ON OFF)
sens.09.useDefault	CHOICE		EnumRange (ON OFF)
sens.10.useDefault	CHOICE		EnumRange (ON OFF)
sens.11.useDefault	CHOICE		EnumRange (ON OFF)
sens.12.useDefault	CHOICE		EnumRange (ON OFF)
sens.13.useDefault	CHOICE		EnumRange (ON OFF)
sens.14.useDefault	CHOICE		EnumRange (ON OFF)
sens.15.useDefault	CHOICE		EnumRange (ON OFF)
sens.16.useDefault	CHOICE		EnumRange (ON OFF)
internal.meas	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.max	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.01.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.02.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.03.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.04.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)

name	*type*	*flags*	*range*
internal.05.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.06.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.07.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.08.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.09.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.10.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.11.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.12.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.13.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.14.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.15.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)
internal.16.unit	CHOICE	R/O	EnumRange R/O (NONE OTHER V A W VA Wh VAh °C Hz % m/s Pa psi g °F ft inch cm m rpm ° lx gr/cm2 VAr)

name	*type*	*flags*	*range*
internal.01.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.02.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.03.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.04.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.05.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.06.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.07.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.08.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.09.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.10.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.11.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.12.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.13.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.14.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.15.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.16.scale	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 General)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 General)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 General)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 General)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 General)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 General)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 General)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 General)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 General)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 General)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 General)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 General)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 General)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 General)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 General)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 General)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 Low Warn)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 High Warn)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 Low)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Sensor 1 High)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 Low Warn)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 High Warn)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 Low)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Sensor 2 High)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 Low Warn)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 High Warn)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 Low)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Sensor 3 High)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 Low Warn)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 High Warn)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 Low)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Sensor 4 High)

name	*type*	*flags*	*range*
faults.33	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 Low Warn)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 High Warn)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 Low)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Sensor 5 High)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 Low Warn)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 High Warn)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 Low)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Sensor 6 High)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 Low Warn)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 High Warn)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 Low)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Sensor 7 High)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 Low Warn)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 High Warn)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 Low)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Sensor 8 High)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 Low Warn)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 High Warn)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 Low)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Sensor 9 High)

name	*type*	*flags*	*range*
faults.53	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 Low Warn)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 High Warn)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 Low)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Sensor 10 High)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 Low Warn)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 High Warn)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 Low)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Sensor 11 High)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 Low Warn)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 High Warn)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 Low)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Sensor 12 High)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 Low Warn)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 High Warn)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 Low)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Sensor 13 High)

name	*type*	*flags*	*range*
faults.69	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 Low Warn)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 High Warn)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 Low)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Sensor 14 High)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 Low Warn)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 High Warn)
faults.75	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 Low)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Sensor 15 High)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 Low Warn)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 High Warn)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 Low)
faults.80	ALARM	R/O	AlarmFlagRange R/O (Sensor 16 High)

9.570 Remote-Alarm

The Remote-Alarm logical device is part of the remote alarm client feature of the software which reports failures of a M&C system to a remote site through a ('on demand') dial up connection.

If you launch a client (user interface program) to a M&C computer from a remote site through a 'dial on demand' connection, the watchdog mechanism built in the software keeps this connection open, even if there is no other data exchange. May be you simply want to be informed at the remote site if the M&C system shows a fault. In this case the Remote-Alarm facility of the software helps you to keep the communication costs as low as possible.

To make this feature work, configure a Remote-Alarm logical device at the M&C server and run the alarm client program at the remote site. Normally there will be no active network connection between the sites. If, however, the summary fault flag of the M&C system or another one of the configured flags changes, the Remote-Alarm logical device will connect to the remote client program on order to update the signalling there.

Beside signalling the M&C system's summary alarm with a OK/FAULT display on the screen and the keyboard bell, the remote alarm client program is able to switch some relay circuits due to alarm conditions in the M&C system. The Remote-Alarm device in the M&C server maps alarm flags to relay circuits. Controlling the relay circuits is optional, if there is no appropriate hardware at the client machine installed, nothing happens, no error messages appear. The first relay circuit (number 0) always is mapped to the M&C system summary fault. The remaining 5 circuits are user configurable, these assignments are part of the configuration parameters described below.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the Remote-Alarm device into the M&C setup.

parameter	***description***
clientIP	Enter the IP address of the machine running the remote alarm client. If the remote machine's name is known by DNS or listed in the computer's /etc/host file, you may enter the remote machine's name, too.
clientPort	Enter the IP port number where the remote alarm client accepts connections from this machine. The standard value is 2010, you should not change this unless you know exactly which IP ports on the client machine are free. If another port number than 2010 is used, is also must be set as a command line parameter when the remote alarm client is invoked.
faultId1	Enter the message identifier of the fault flag which shall set the relay circuit 1 at the remote client. Leave it empty if the remote client has no relay circuits installed or if you don't want to control this particular relay circuit.
faultId2	Enter the message identifier of the fault flag which shall set the relay circuit 2 at the remote client. Leave it empty if the remote client has no relay circuits installed or if you don't want to control this particular relay circuit.

parameter	*description*
faultId3	Enter the message identifier of the fault flag which shall set the relay circuit 3 at the remote client. Leave it empty if the remote client has no relay circuits installed or if you don't want to control this particular relay circuit.
faultId4	Enter the message identifier of the fault flag which shall set the relay circuit 4 at the remote client. Leave it empty if the remote client has no relay circuits installed or if you don't want to control this particular relay circuit.
faultId5	Enter the message identifier of the fault flag which shall set the relay circuit 5 at the remote client. Leave it empty if the remote client has no relay circuits installed or if you don't want to control this particular relay circuit.

Running the alarm client program

To run the alarm client on a PC with an installed sat-nms client use the same command line but change the parameters after 'satnms.jar': *[jre] -cp satnms.jar satnms.gui.FaultFrame <IP address of M&C system> [Options]*

Possible options are:

- **-s** : silent, no key-bell on error
- **-t title** : defines the window title
- **-l port-number** : defines the local IP port where the alarm client listens for connections. Without this parameter the alarm client uses port 2010.
- **-p COMn** : switch relay circuits of a connected NAIS-FP0 on COM n
- **-d name** : name of the RemoteFault Device which communicates with this alarm client. Default: RMTFLT

Example command line for Windows: *jrew.exe -cp satnms.jar satnms.gui.FaultFrame 192.168.10.10*

Chosing port numbers

For standard applications you should use the default port number 2010 whenever possible. However, there may be situations where it is necessary to use an alternate port number. For example, if you are running two instances of the alarm client on one machine showing the fault state of two particular M&C systems. In this case, the alarm clients must use different port numbers.

You should ensure, that the chosen port number isn't used by another application and you must set the same port number with the **-l port-number** switch at the alarm client and at the Remote-Alarm device's setup window as well.

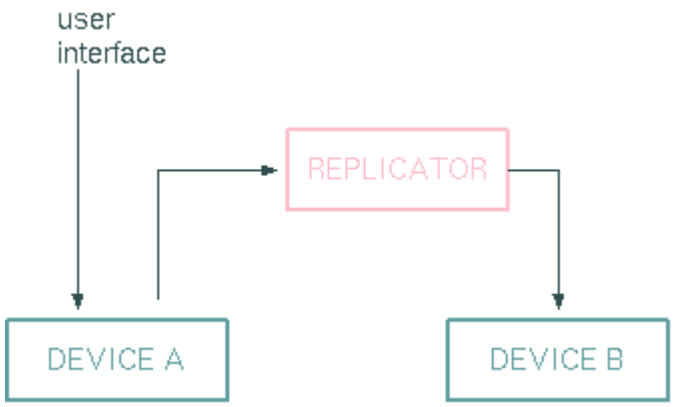
Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.clientIP	TEXT	SAVE SETUP	StringRange
config.clientPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.faultId1	TEXT	SAVE SETUP	StringRange
config.faultId2	TEXT	SAVE SETUP	StringRange
config.faultId3	TEXT	SAVE SETUP	StringRange
config.faultId4	TEXT	SAVE SETUP	StringRange
config.faultId5	TEXT	SAVE SETUP	StringRange
reset	TEXT		StringRange

9.571 Replicator

The Replicator logical device replicates a parameter setting from one device to another. A typical application for the Replicator is to bind frequencies or data rates at communicating devices to the same value.



The Replicator listens to the reply of the primary parameter the operator controls and commands this value to the secondary device. You should only bind parameters of the same type and preferable the same range. You will get unpredictable results when binding a frequency to a switch position e.g.

Filtering

Additionally, the Replicator device is capable to filter the replicated messages. If you set the *filter* configuration parameter to a comma separated list of values, the Replicator device only processes messages which match one of the given values. You should not use this feature with floating point parameters as rounding inaccuracies may cause unpredictable results.

Conditional enabling

The replication function may be enabled / disabled by an external parameter. If the*enableId* configuration parameter is set, the device watches this parameter and replicates only if the parameter value matches the reference value set with the *enableVal* configuration parameter. The *info.enabled* status variable tells if replication is actually enabled.

If the enable state changes from false to true, the device replicates the input once - provided the actual input value matches the filter. If *enableId* ist empty or undefined, the replication function is always enabled.

Startup behavior

After system start the device stays disabled for 10 driver cycles (which are about 10 seconds at an IDLE time of 1000msecs). At the end of this initialization delay it replicates the input value once if all enabling factors are true. This avoids unintended replications during the startup phase if the first input value arrives before filter and enableId are processed by the device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Replicator device with the devices that will perform the frequency setting.

parameter	***description***
inputId	Enter the message identifier here of the parameter the replicator shall listen to, e.g. "MODULATOR.tx.dataRate".
outputId	Enter the message identifier here of the parameter the replicator shall control, e.g. "ENCODER.dataRate".

parameter	*description*
filter	Enter a comma separated list of the values that the Replicator device shall process. If a 'filter' list is given, the Replicator ignores all messages which do not match one of the values of the filter list. Leave the list empty to prevent the Replicator from performing any filtering (all messages are replicated in this case).
enableId	If set, the device watches this parameter and enables replication only if the value of this parameter matches the *enableVal* defined below.
enableVal	The value to compared to the value of the enabling parameter defined above with *enableId*. The device does a string compare with the value and its reference, hence comparing floating point values will give unpredictable results.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
info.enabled	TEXT	R/O	StringRange R/O
config.inputId	TEXT	SAVE SETUP	StringRange
config.outputId	TEXT	SAVE SETUP	StringRange
config.filter	TEXT	SAVE SETUP	StringRange
config.enableId	TEXT	SAVE SETUP	StringRange
config.enableVal	TEXT	SAVE SETUP	StringRange

9.572 Report-Generator

The Report-Generator logical device collects a user defined set of parameters and generates formatted reports on a regular basis. Reports are stored as PDF files for printout and in parallel as semicolon separated text files (CSV) for further machine based processing. The Report-

Generator device is available as of version 3.0 of the software.

Report types

The Report-Generator provides a number of report types. Each type implements a certain report generation timing. The following table shows the available report types:

- **SIMPLE** --- The simple report type provides a freely definable update interval. It produces a single column report every *n* days. Additionally, this report type is the only one which allows to set the time of day when the report shall be created.
- **WEEKLY** --- The weekly report creates report sheets with 7 columns. The report sheet gets updated every day at midnight, a new report file is started every Monday 00:00 UTC
- **MONTHLY** --- The monthly report creates a 31-column table. Each report file starts at the 1st of a month at 00:00 UTC, the table gets updated on a daily basis. A 31-column table on a single page would lead to very narrow columns and a very small font size. The table therefore is spread over two pages, one for the 1st .. 16th of the month, the second for the 17th to the end of month. Page and table headers are repeated accordingly.
- **QUARTERLY** --- The quarterly report collects the data for one quarter on a report sheet, updated once a month (three columns)
- **YEARLY** --- The yearly report collects the data over one year, arranged on 12 columns, one for each month.

The report timing (the report type and for simple reports also the interval and the time of day) is set at the device window of the Report-Generator device. Please note that changing the report timing voids all temporary data collected for the actual report interval.

Report contents

The contents of a report produced by the Report-Generator device is defined in a text based configuration file. The file defines which parameters are to be recorded and also some information about the layout of the report sheets.

The configuration file defines one report property or report item per line. Empty lines and lines starting with a hash character ('#') are ignored. Each line starts with a keyword, followed by a (item specific) number of parameters.

- **VAL** --- Adds a standard item to the report. The report contains the parameter value at the end of the recorded interval. VAL must be followed by the parameter identifier to record and a (mandatory) description.
  - Example:**VAL DEVICE-1.param2 This is the descriptive text**
- **MAX** --- Adds a 'maximum' item to the report. The report contains the maximum value found over the recorded interval. MAX must be followed by the parameter identifier to record and a (mandatory) description.
  - Example:**MAX DEVICE-1.param2 This records a maximum value** Using MAX with non-numeric parameters gives no meaningful results.
- **MIN** --- Adds a 'minimum' item to the report. The report contains the minimum value found over the recorded interval. MIN must be followed by the parameter identifier to record and a (mandatory) description.
  - Example:**MIN DEVICE-1.param2 This records a minimum value** Using MIN with

non-numeric parameters gives no meaningful results.

- **AVG** --- Adds an 'average' item to the report. The report contains the average computed over the recorded interval. AVG must be followed by the parameter identifier to record and a (mandatory) description.
  - Example:**AVG DEVICE-1.param2 This records an average value** Using AVG with non-numeric parameters gives no meaningful results.
- **EQU** --- Adds an 'equ' item to the report. This item collects the time while the given parameter equals a reference value. EQU must be followed by the parameter identifier to record, the reference value and a (mandatory) description.
  - Example:**EQU DEVICE-1.param2 REFVAL Records the time while param2 equals REFVAL**
- **NEQ** --- Adds an 'neq' item to the report. This item collects the time while the given parameter does not equal a reference value. NEQ must be followed by the parameter identifier to record, the reference value and a (mandatory) description.
  - Example:**NEQ DEVICE-1.param2 REFVAL Records the time while param2 does not equal REFVAL**
- **LML** --- Adds a low limit item to the report. This item collects the time while the given parameter falls below a limit value. LML must be followed by the parameter identifier to record, the limit value and a (mandatory) description.
  - Example:**LML DEVICE-1.param2 -77.5 Records the time while param2 is below the limit**Using LML with non-numeric parameters gives no meaningful results.
- **LMH** --- Adds a high limit item to the report. This item collects the time while the given parameter is above a limit value. LMH must be followed by the parameter identifier to record, the limit value and a (mandatory) description.
  - Example:**LMH DEVICE-1.param2 45 Records the time while param2 is above the limit**Using LMH with non-numeric parameters gives no meaningful results.
- **PAUSED** --- Adds a special item to the report which records the time while the Report-Generator was in PAUSED state. PAUSED is followed by descriptive text.
  - Example:**PAUSED Maintenance time / recording paused**
- **SPACE** --- Adds an empty line to the report. There are no parameters following this keyword.
- **TEXT** --- Adds a report line with only the description column filled. TEXT is followed by the descriptive text to print.
  - Example:**TEXT Example text**
- **HEADING** --- Heading works similar to TEXT, but the text is bold printed and may cover the whole report line.
  - Example:**HEADING Bold printed text**
- **LEFT-TITLE** --- Each report sheet contains a caption/title at it's top. LEFT-TITLE supplies the text for the left title field.
  - Example:**LEFT-TITLE title text**
- **MID-TITLE** --- MID-TITLE supplies the text for the center title field.
  - Example:**MID-TITLE title text**
- **RIGHT-TITLE** --- RIGHT-TITLE supplies the text for the right title field.
  - Example:**RIGHT-TITLE title text**

The sequence of item definitions in the configuration file defines the sequence of lines in the

report. The global TITLE definitions may appear anywhere in the file.

When using non-ASCII characters in the description texts, the file must be ISO8859 encoded that these characters are correctly shown in the report.

Snapshot reports

The Report-Generator device is capable to produce snapshot reports on a button press. The snapshot report prints the actual values of the recorded parameters. Report items which record time intervals are omitted in this report. Minimum/maximum/average items print the actual parameter value rather than the value computer over a certain time interval.

Pausing the data recording

The device allows to pause data recording by setting the appropriate parameter at the device window's main page. While paused, the device does not recognize any changes of parameter values. This may be helpful during maintenance of the station: Faults of incorrect parameter readings produced during the exchange of a device do not falsify the report statistics if the Report-Generator is set to PAUSED during the maintenance operation.

In order to remind the operator of re-enabling the device, the Report-Generator shows a fault while it is PAUSED.

Report file names

The device uses a fixed pattern to generate the names of the report files. Each file name starts with the name of the Report-Generator device, followed by a single letter for the report type and the description of the recorded time interval. Examples:

- **FOO-W-14-2010.pdf** --- A weekly report for week 12 in 2010, generated by the device FOO
- **FOO-M-05-2010.pdf** --- A monthly report for May 2010, generated by the device FOO
- **FOO-Q-03-2010.pdf** --- A quarterly report for the third quarter 2010 (July - September), generated by the device FOO
- **FOO-Y-2010.pdf** --- A yearly report for 2010, generated by the device FOO
- **FOO-S-2010-04-22.pdf** --- A simple report, starting on April, 22th 2010, generated by the device FOO

Snapshot reports do not include a type letter, their file names consists of the device name and a full time stamp of the report creation.

CSV report files

Beside the PDF formatted reports, the device automatically produces CSV files for further machine based processing of the reported data. The files contain a semicolon separated table with the same layout as in the corresponding PDF report, except for TEXT, SPACE and HEADING items which are omitted in the CSV file. Also the CSV files are not page-formatted and do not contain page headers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The settings page shows and defines the timing parameters for the report.
- --- The state page shows a summary of the data, the device actually watches.
- --- The device faults page. The device rises a fault if it could not read the configuration file and also while data recording is paused.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the device work properly.

parameter	***description***
configFile	The name of the configuration file which defines the report content.
reportDirectory	The name of the directory where the report files shall be stored. This may be an absolute path or a path relative to the software's working directory. The directory must exist, the software does not create it automatically.

Additional remarks

The Report-Generator device stores all temporary data needed to collect the data for a report - including the timing settings for the report - in a file called XXX.dat, where XXX is substituted by the name of the device. This backup file is updated once an hour. The file is located in the base installation directory of the sat-nms software. Deleting this file resets the Report-Generator device to a daily SIMPLE report type and voids all cached report data.

The configuration file defining the report contents may be changed "on the fly", e.g. to add new parameters. After changing the file, press "RESET" on the maintenance page at the Report-Generator device window. This makes the device re-read the file. The collected data of all item which match in item type and parameter identifier is retained. New or changed report items are newly initialized.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.configFile	TEXT	SAVE SETUP	StringRange
config.reportDirectory	TEXT	SAVE SETUP	StringRange
reportType	CHOICE		EnumRange (SIMPLE WEEKLY MONTHLY QUARTERLY YEARLY)
recording	CHOICE		EnumRange (RUNNING PAUSED)
timeOfDay	TEXT	SAVE	StringRange
interval	INTEGER	SAVE	IntegerRange (1 .. 365)
snapshot	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Recording stopped)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

9.573 ResearchConcepts-RC1500

This page describes the device driver and the device window for the Research Concepts RC1500 single axis / polarmount antenna control unit.

The drivers was tested with model RC1500A and does not include support for tracking, AGC and AutoPol features and you cannot use the included satellite positions (targets). Please use the sat-nms Antenna Pointing logical device to store and recall satellite positions.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

parameter	*description*
address	This is the only parameter that must be set. Factory default for the serial remote control port is 50.
jogduration	You can configure the jog duration of the antenna controller. This allows you to define, how long the antenna is moving if you press one of the arrow-buttons on the main pointing page.
polAdjust	Set to ON if you want to control the polarization. See chapter*Polarization* above.
posLoLimit	Position low limit (ccw). This is only a software limit and not the limit stored while the RC1500 setup procedure. Use a value which is greater then the internal limit of the controller.
posHiLimit	Position high limit (cw). This is only a software limit and not the limit stored while the RC1500 setup procedure. Use a value which is smaller then the internal limit of the controller.
plLoLimit	Polarization low limit (ccw). This is only a software limit and not the limit stored while the RC1500 setup procedure. Use a value which is greater then the internal limit of the controller.
plHiLimit	Polarization high limit (cw). This is only a software limit and not the limit stored while the RC1500 setup procedure. Use a value which is smaller then the internal limit of the controller.
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,7,E,1* .

Remarks

Polarization

If you have a movable polarization, you have to set the setup parameter polAdjust to ON. This enables the required parameters. The drivers detects if you have an Polarrotor or a Rotating Feed installed (option RC2KPOL). Regarding this you have different ranges for the polarization. The polAdjust parameter on the Info page will show you the detected type.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.polAdjust	TEXT	R/O	StringRange R/O
internal.state.pos	TEXT	R/O	StringRange R/O
internal.state.pl	TEXT	R/O	StringRange R/O
internal.state.pl2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.rpn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.test	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.jogduration	TEXT	SAVE SETUP	StringRange
config.polAdjust	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.posLoLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 65635)
config.posHiLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 65635)
config.plLoLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1023)
config.plHiLimit	INTEGER	SAVE SETUP	IntegerRange (0 .. 1023)
position	INTEGER		IntegerRange (0 .. 65535)
polarization	FLOAT		DoubleRange DIS (0 .. 0)
step	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
target.name	TEXT	R/O	StringRange R/O
state.pos	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
state.pl	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.plui	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pos.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.cw	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
state.limit.pos.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.ccw	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
state.moving.pos	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.pl	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Pos runaway)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Pos jammed)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Pos limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Pos drive)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Pol limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Pol alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Pol options)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Options)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low batterie)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Antenna)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Count)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Limits corrupt)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Autopol cfg)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slow speed)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (Com port)

9.574 ResearchConcepts-RC2000

This page describes the device driver and the device window for the Research Concepts RC2000 two axis antenna control unit.

The drivers was tested with model 2KCA (elevation over azimuth mounts) with Polarization adjustment via rotating feed and does not include support for tracking, AGC and AutoPol features and you cannot use the included satellite positions (targets). Please use the sat-nms Antenna Pointing logical device to store and recall satellite positions.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

parameter	*description*
address	This is the only parameter that must be set. Factory default for the serial remote control port is 50.
jogduration	You can configure the jog duration of the antenna controller. This allows you to define, how long the antenna is moving if you press one of the arrow-buttons on the main pointing page.
azA,azB,azC	Parameters of the polynomial to convert the ticks from an actuator for azimuth to degrees (see description above)
elA,elB,elC	Parameters of the polynomial to convert the ticks from an actuator for elevation to degrees (see description above)
plFactor	Converts the ticks from an actuator for polarization to degrees (see description above)

parameter	***description***
plOffset	Converts the ticks from an actuator for polarization to degrees (see description above)
plRevOffset	Converts the ticks from an actuator for polarization to degrees (see description above)
Serial	Remember to set the line parameters for the serial interface connected to the up-converter. It is recommended to operate the device at *9600,7,E,1* .

Calculating degrees for Azimuth and Elevation

To calculate the degrees from the ticks the driver uses the following formula:

- *azimuthInDegrees = azC + azB* azimuthInTicks + azA * azimuthInTick^2*

These values depends on the actuators and the your antenna.

Steps to determine the factors and offsets

You need to approximate the following formula:

- y=a*x^2+b*x+c

With the support of an spread sheet application this is not so difficult:

- Point (and optimize) your antenna to (all) satellite positions you want to use and write down the "ticks" from the antenna controller for azimuth, elevation and polarization
- Calculate the theoretical values for azimuth and elevation with the help of the satellite orbit position and you geo-location.
- Create diagrams (one for azimuth, one for elevation and one for polarization which have the "ticks" on the x-axis and the calculated degrees on the y-axis.
- Create a trend line with the spreadsheet application - type polynomial, Level: 2 - and select the option "show formula inside the diagram". For polarization use type: linear
- Take the values for a (azA|elA), b (azB|elB)and c (azC|elC) from the formula and insert it into the device setup of the antenna controller device. For polarization you need only b (plFactor) and c (plOffset) and calculate plRevOffset = plOffset / plFactor
- Check if you can point to all your satellite position.

Calculating degrees for Polarization

To calculate the degrees from the ticks the driver uses the following formula:

- *azimuthInDegrees = (config.azFactor * azimuthInTicks) + config.azOffset*

These values depends on the actuators and the your antenna.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pl2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state.az	TEXT	R/O	StringRange R/O
internal.state.el	TEXT	R/O	StringRange R/O
internal.state.pl	TEXT	R/O	StringRange R/O
internal.state.pl2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.rpn	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.jogduration	TEXT	SAVE SETUP	StringRange
config.azA	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.azB	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.azC	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.elA	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.elB	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.elC	FLOAT	SAVE SETUP	DoubleRange (0.0000000E+00 .. 0.0000000E+00)
config.plFactor	FLOAT	SAVE SETUP	DoubleRange (0.000000 .. 0.000000)
config.plOffset	FLOAT	SAVE SETUP	DoubleRange (0.000000 .. 0.000000)
config.plRevOffset	FLOAT	SAVE SETUP	DoubleRange (0.000000 .. 0.000000)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
step	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.el	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.az.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.up	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.az.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.down	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.az	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.el	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving.pl	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
internal.polRequested	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Az runaway)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Az jammed)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Az limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Az drive)
faults.05	ALARM	R/O	AlarmFlagRange R/O (El runaway)
faults.06	ALARM	R/O	AlarmFlagRange R/O (El jammed)
faults.07	ALARM	R/O	AlarmFlagRange R/O (El limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (El drive)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low batterie)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Az alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (El alarm)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Az count)
faults.13	ALARM	R/O	AlarmFlagRange R/O (El count)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Az limit corrupt)
faults.15	ALARM	R/O	AlarmFlagRange R/O (El limit corrupt)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Simultaneous Az/El)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Az slow speed)
faults.18	ALARM	R/O	AlarmFlagRange R/O (El slow speed)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Com port data)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Time or Date)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Az/El ang display)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Pol options)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Pol alarm)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Az/El options)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Pol limit)

9.575 ResearchConcepts-RC2500

This page describes the device driver and the device window for the Research Concepts RC2500 antenna control unit. The driver directly controls units which implement all three axes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving. If an target was selected the name of the target will be shown.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

You can configure the jog duration of the antenna controller. This allows you to define, how long the antenna is moving if you press one of the arrow-buttons on the main pointing page.

parameter	***description***
address	This is the only parameter that must be set. Factory default for the RS232 remote control port is 65.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
AGCChannel	Defines the AGC channel NONE/RF/SS1/SS2
jogduration	Defines the time in msec for a jog step.
jogduration.pl	Defines the time in msec for a polarization jog step.
azOffset	Defines an offset if necessary to match the azimuth angle to the true azimuth value.
elOffset	Defines an offset if necessary to match the elevation angle to the true elevation value.
plOffset	Defines an offset if necessary to match the polarization angle to the true polarization value.
azLoLimit	Defines the lower/ccw azimuth limit.
azHiLimit	Defines the upper/cw azimuth limit.
elLoLimit	Defines the lower/ccw elevation limit.
elHiLimit	Defines the upper elevation limit.
plLoLimit	Defines the lower/ccw polarization limit.
plHiLimit	Defines the upper/cw polarization limit.

Remarks

The RC2500 must have a firmware version with remote control enabled.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.rotatingFeed	TEXT	R/O	StringRange R/O
config.readbackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.AGCChannel	CHOICE	SETUP	EnumRange (none RF SS1 SS2)
config.jogduration	TEXT	SAVE SETUP	StringRange
config.jogduration.pl	TEXT	SAVE SETUP	StringRange
config.azOffset	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.elOffset	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.plOffset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
step	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
updateDest	TEXT	NOPRESET	StringRange
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.limit.az.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.up	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.az.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.down	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
speed.az	CHOICE	R/O	EnumRange R/O (FAST SLOW)
speed.el	CHOICE	R/O	EnumRange R/O (FAST SLOW)
speed.pl	CHOICE	R/O	EnumRange R/O (FAST SLOW)
state.moving.az	CHOICE	R/O	EnumRange R/O (UNKNOWN STOP PENDING CW CCW AUTO)

name	*type*	*flags*	*range*
state.moving.el	CHOICE	R/O	EnumRange R/O (UNKNOWN STOP PENDING UP DOWN AUTO)
state.moving.pl	CHOICE	R/O	EnumRange R/O (UNKNOWN STOP PENDING CW CCW AUTO)
motorStop	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
info.state.az	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
info.state.el	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
info.state.pl	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
internal.pl	INTEGER	R/O	IntegerRange R/O (0 .. 65535)
internal.azf	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elf	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plf	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.limit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.polRequested	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Az sensor direction)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Az runaway alarm)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Az jammed alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Az drive alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (El sensor direction)
faults.06	ALARM	R/O	AlarmFlagRange R/O (El runaway alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (El jammed alarm)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (El drive alarm)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Pol sensor direction)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Pol runaway alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Pol jammed alarm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Pol drive alarm)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Low batterie)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Azimuth)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Elevation)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Outdoor mintenance)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Outdoor limit)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Inclined orbit data)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Antenna/Recv. data)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Time/Date)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Offsets/Sensors)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Polarization data)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Polarization)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Auto move data)

9.576 ResearchConcepts-RC3000

This page describes the device driver and the device window for the Research Concepts RC3000 antenna control unit. The driver directly controls units which implement all three axes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving. If an target was selected the name of the target will be shown.
- --- This page implements the LOCATE command of the RC3000
- --- This page implements the RECALL command of the RC3000
- --- This page implements the STORE command of the RC3000
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

You can configure the jog duration of the antenna controller. This allows you to define, how long the antenna is moving if you press one of the arrow-buttons on the main pointing page.

parameter	***description***
address	This is the only parameter that must be set. Factory default for the RS232 remote control port is 50.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
AGCChannel	Defines the AGC channel.
jogduration	Defines the time in msec for a jog step.
azOffset	Defines an azimuth offset if necessary to match the azimuth angle to the true azimuth value.
azLoLimit	Defines the lower/ccw azimuth limit.
azHiLimit	Defines the upper/cw azimuth limit.
elLoLimit	Defines the lower/ccw elevation limit.

parameter	*description*
elHiLimit	Defines the upper elevation limit.
plLoLimit	Defines the lower/ccw polarization limit.
plHiLimit	Defines the upper/cw polarization limit.

Remarks

- The RC3000 must have a firmware version with remote control enabled. This is indicated by the last letter in the version string e.g. RC3K-S1-GNR which must be a R.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.rotatingFeed	TEXT	R/O	StringRange R/O
internal.azLimit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.elLimit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.plLimit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.azStatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.elStatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.plStatus	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.az	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.test	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.line1	TEXT	R/O	StringRange R/O
internal.line2	TEXT	R/O	StringRange R/O
internal.line3	TEXT	R/O	StringRange R/O
internal.line4	TEXT	R/O	StringRange R/O
internal.polRequested	BOOLEAN	R/O	BooleanRange R/O (true false)
test	TEXT		StringRange
config.readbackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.AGCChannel	CHOICE	SETUP	EnumRange (none RF SS1 SS2)
config.jogduration	TEXT	SAVE SETUP	StringRange
config.jogduration.pl	TEXT	SAVE SETUP	StringRange
config.azOffset	FLOAT	SAVE SETUP	DoubleRange (-360.0 .. 360.0)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.0 .. 0.0)
elevation	FLOAT		DoubleRange (0.0 .. 0.0)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
updateDest	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
recall.longitude	FLOAT	R/O	DoubleRange R/O (-179.9 .. 179.9)
recall.inclination	INTEGER	R/O	IntegerRange R/O (0 .. 19)
recall.rfband	CHOICE	R/O	EnumRange R/O (C-Band Ku-Band C/Ku-Band L-Band X-Band Ka-Band S-Band)
recall.poloffset	FLOAT	R/O	DoubleRange R/O (-90.0 .. 90.0)
recall.satname	TEXT	R/O	StringRange R/O
recall.polselect	CHOICE		EnumRange (H V N)
recall.posupdate	CHOICE		EnumRange (auto force)
store.longitude	FLOAT		DoubleRange (-179.9 .. 179.9)
store.inclination	INTEGER		IntegerRange (0 .. 19)
store.rfband	CHOICE		EnumRange (C-Band Ku-Band C/Ku-Band L-Band X-Band Ka-Band S-Band)
store.poloffset	FLOAT		DoubleRange (-90.0 .. 90.0)
store.satname	TEXT		StringRange
store.polselect	CHOICE		EnumRange (H V N)
store.posupdate	CHOICE		EnumRange (auto force)
longitude	FLOAT		DoubleRange (-179.9 .. 179.9)
inclination	INTEGER		IntegerRange (0 .. 19)
rfband	CHOICE		EnumRange (C-Band Ku-Band C/Ku-Band L-Band X-Band Ka-Band S-Band)
poloffset	FLOAT		DoubleRange (-90.0 .. 90.0)
polselect	CHOICE		EnumRange (H V N)
posupdate	CHOICE		EnumRange (auto force)
internal.namelong	TEXT		StringRange
txpolplane	CHOICE		EnumRange (N H V h v)
step	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange

name	*type*	*flags*	*range*
state.az	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.el	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.stow.az	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stow.el	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stow.pl	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon	TEXT	R/O	StringRange R/O
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.limit.az.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.up	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.cw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.az.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.el.down	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit.pl.ccw	BOOLEAN	R/O	BooleanRange R/O (true false)
state.display	TEXT	R/O	StringRange R/O
key	TEXT	NOPRESET	StringRange
speed.az	CHOICE		EnumRange (fast slow)
speed.el	CHOICE		EnumRange (fast slow)
speed.pl	CHOICE		EnumRange (fast slow)
state.moving.az	CHOICE		EnumRange (not moving cw ccw auto)
state.moving.el	CHOICE		EnumRange (not moving up down auto)
state.moving.pl	CHOICE		EnumRange (not moving cw ccw auto)
clear	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
stow	TEXT	NOPRESET	StringRange
deploy	TEXT	NOPRESET	StringRange
locate	TEXT	NOPRESET	StringRange
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 23)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 23)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Az sensor direction)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Az runaway alarm)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Az jammed alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Az drive alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (El sensor direction)
faults.06	ALARM	R/O	AlarmFlagRange R/O (El runaway alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (El jammed alarm)
faults.08	ALARM	R/O	AlarmFlagRange R/O (El drive alarm)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Pol sensor direction)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Pol runaway alarm)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Pol jammed alarm)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Pol drive alarm)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Low batterie)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Communication port)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Track configuration)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Time/Date)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Limits inactive warning)

9.577 Reset-Timer

The Reset-Timer logical device forces an arbitrary parameter to return to a default value after a

given time.

The Reset-Timer device watches the parameter to control. If the actual value of the parameter differs from the given default value, the timer starts running. When the delay time has elapsed, the parameter gets commanded back its default value by the Reset-Timer device.

You can use the Reset-Timer device to ensure that a certain parameter automatically returns to a given default value after some time.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The timer parameters and its state.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Reset-Timer device with the parameter to control.

parameter	***description***
parameterId	The ID of the parameter to control
defaultValue	The default value of the parameter. The timer will start if the actual parameter value differs from the default value, after the timer has finished, this value will be set at 'parameterId'

Remarks

The Reset-Timer takes care, that it does not start operating until it has received all necessary parameters at startup. However, if you change the configuration parameter of the timer on the fly, the default value and the actual parameter value may differ while you are changing the configuration settings. To prevent the timer from being started in this case, it is recommended to disable the timer while modifying its configuration settings.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.parameterId	TEXT	SAVE SETUP	StringRange
config.defaultValue	TEXT	SAVE SETUP	StringRange
delayTime	INTEGER	SAVE	IntegerRange (0 .. 0)
remainingTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
timerState	TEXT	R/O	StringRange R/O
timerEnable	CHOICE	SAVE	EnumRange (ENABLED DISABLED)

9.578 Romantis-UHP

This page describes the device driver and the device window for the Romantis/Eastar UHP VSAT/TDMA Modem/Terminal.

This is a driver to monitor the state of a hub station and up to 8 remotes.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.cpuLoad	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.isHub	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.remotes	INTEGER	SAVE SETUP	IntegerRange (0 .. 8)
config.00.name	TEXT	SAVE SETUP	StringRange
config.01.name	TEXT	SAVE SETUP	StringRange
config.02.name	TEXT	SAVE SETUP	StringRange
config.03.name	TEXT	SAVE SETUP	StringRange
config.04.name	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.05.name	TEXT	SAVE SETUP	StringRange
config.06.name	TEXT	SAVE SETUP	StringRange
config.07.name	TEXT	SAVE SETUP	StringRange
config.08.name	TEXT	SAVE SETUP	StringRange
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE	SAVE	EnumRange (OFF ON)
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.meter.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
remote.01.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.01.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.01.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.02.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.02.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.02.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.03.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.03.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.03.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.04.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.04.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.04.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
remote.04.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.05.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.05.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.05.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.06.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.06.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.06.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.07.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.07.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.07.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.08.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.08.cton	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
remote.08.crc	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.last	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
clear	TEXT	NOPRESET	StringRange

9.579 RuS-FPH

This page describes the *Rohde&Schwarz FPH Series* device driver and the device window for this device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum

view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

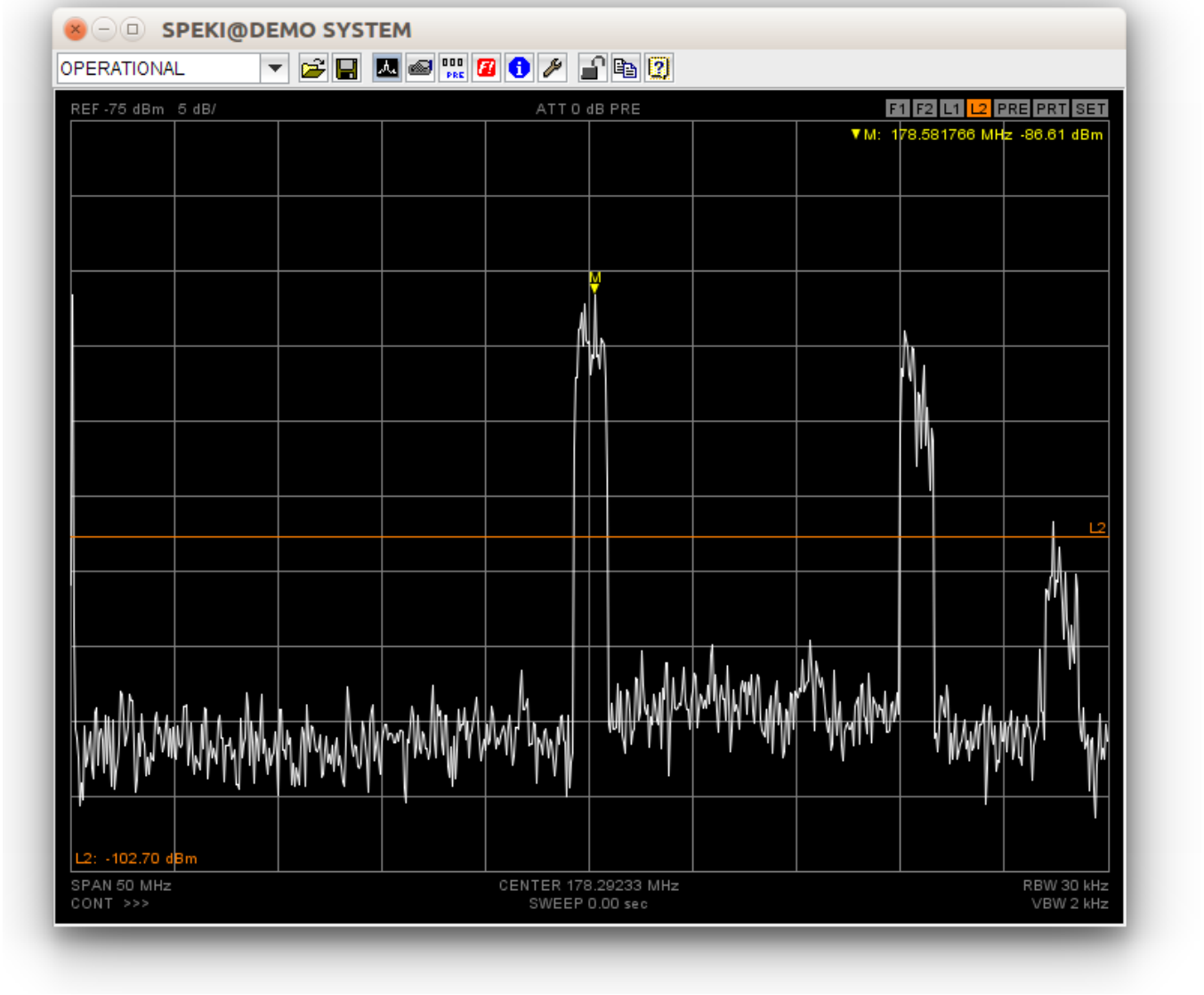
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Rohde & Schwarz FSL driver provides save and load presets function. The Presets are save directly on the driver.

How to configure the driver

The Rohde & Schwarz FSL are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.options	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.inputSelId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.001 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS

name	*type*	*flags*	*range*
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.580 RuS-FSC

This page describes the *Rohde&Schwarz FSC Series* device driver and the device window for this device.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

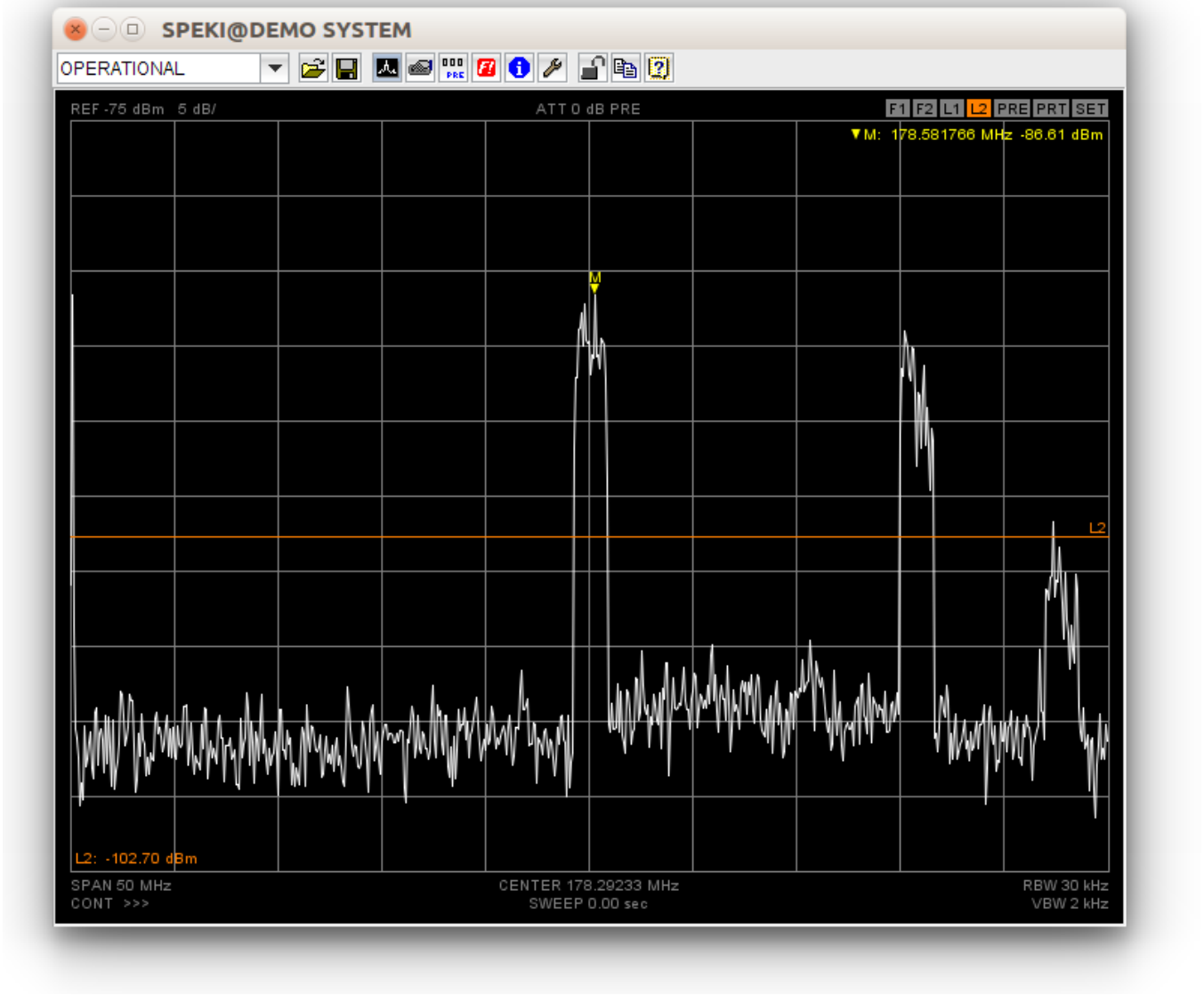
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Rohde & Schwarz FSL driver provides save and load presets function. The Presets are save directly on the driver.

How to configure the driver

The Rohde & Schwarz FSL are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.options	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.inputSelId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.001 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS

name	*type*	*flags*	*range*
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hasB22	BOOLEAN	R/O	BooleanRange R/O (true false)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.581 RuS-FSH

This page describes the *Rohde&Schwarz FSH Series* device driver and the device window for this device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

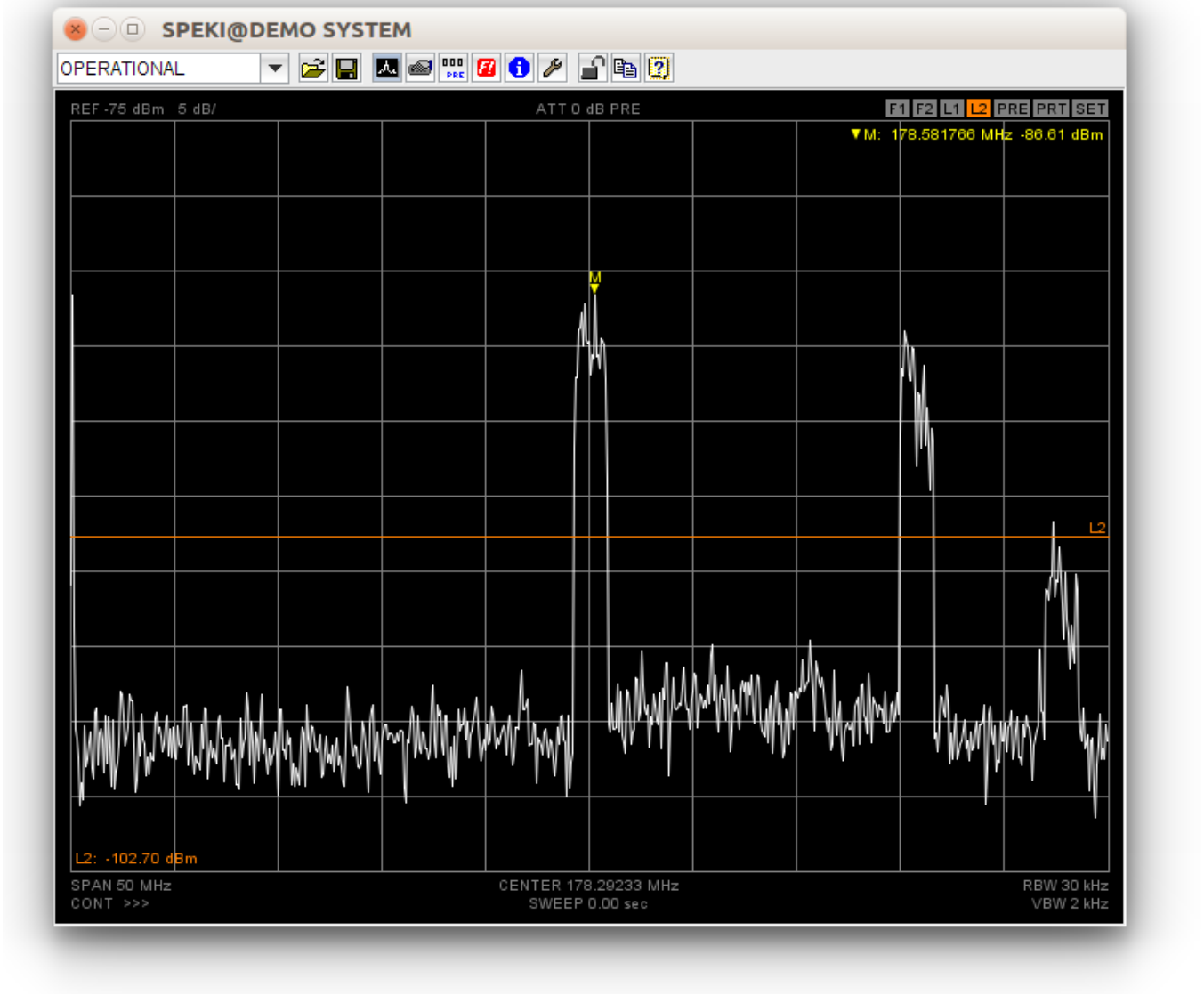
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the

instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Rohde & Schwarz FSL driver provides save and load presets function. The Presets are save directly on the driver.

How to configure the driver

The Rohde & Schwarz FSL are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.options	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.inputSelId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.001 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS

name	*type*	*flags*	*range*
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.582 RuS-FSL

This page describes the *Rohde&Schwarz FSL Series* device driver and the device window for this device.

The drivers was tested with following equipments:

- Model: FSL-3

- SW version: 2.50

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

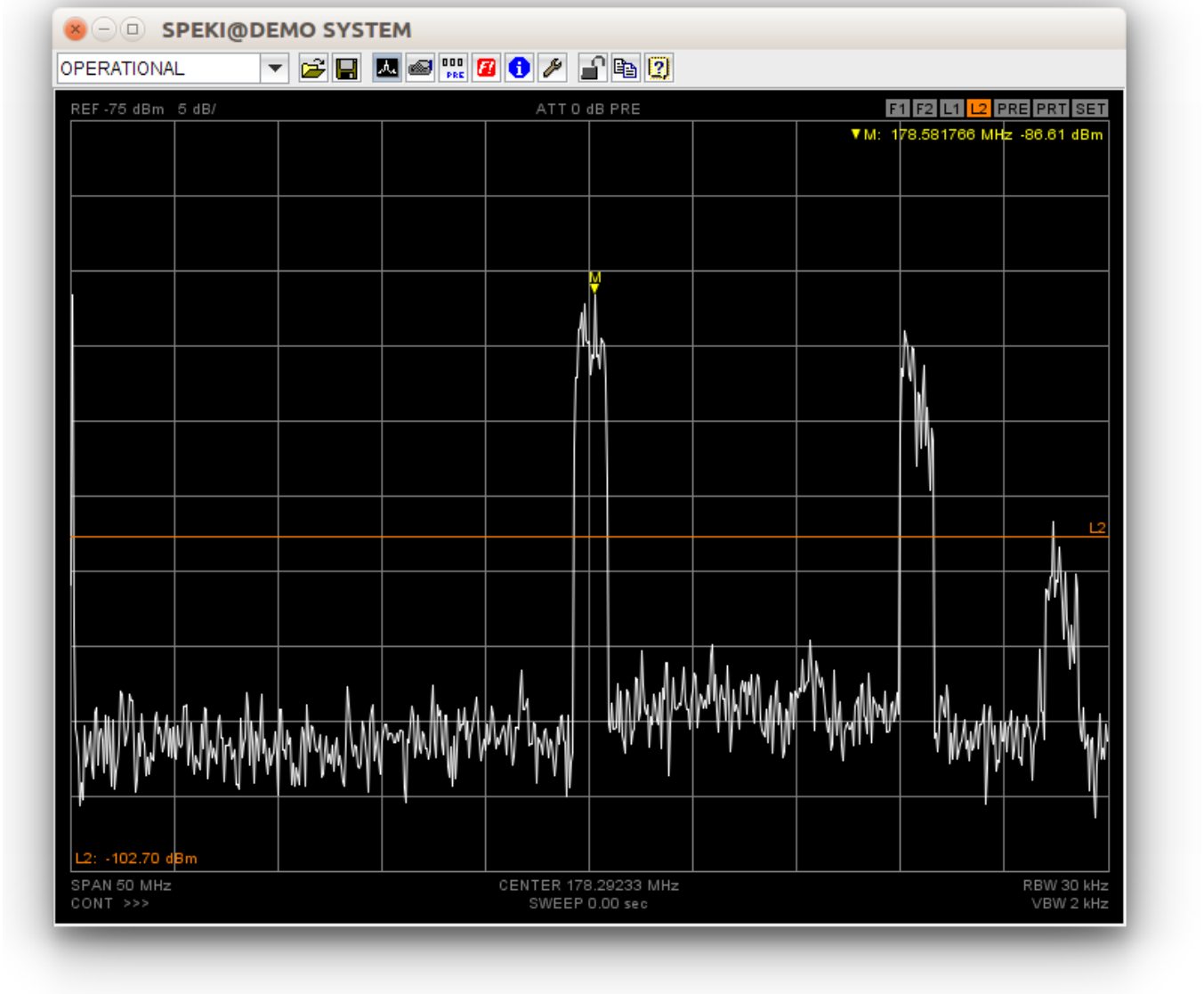
At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and

a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).
- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG

mode is followed by the number of swees already averaged.

Preset handling

The Rohde & Schwarz FSL driver provides save and load presets function. The Presets are save directly on the driver.

How to configure the driver

The Rohde & Schwarz FSL are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.options	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.001 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)

name	*type*	*flags*	*range*
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hasB22	BOOLEAN	R/O	BooleanRange R/O (true false)
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)

9.583 RuS-FSV

This page describes the *Rohde&Schwarz FSV Series* device driver and the device window for this device.

The drivers was tested with following equipments:

- Model: FSV-30
- SW version: 3.70

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The spectrum view.
- --- The main parameter page displays/controls the spectrum main analyzer settings like center frequency, span, etc. You get this page in it's own window beside the spectrum view if you click to 'SET' at the upper left corner of the spectrum view window.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

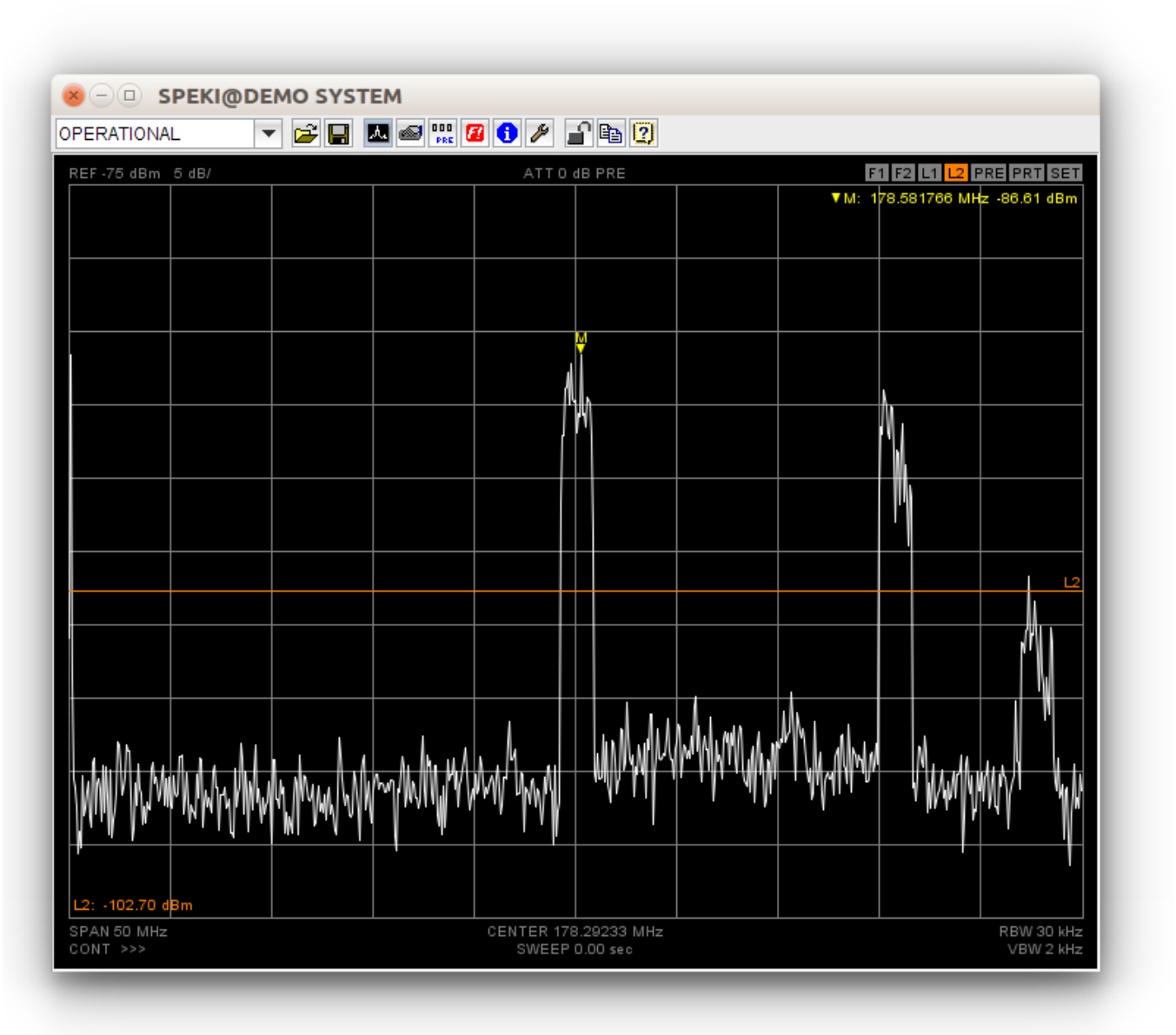
At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
inputSelId	The device driver is capable to monitor the position of a switch or signal matrix output which is used as an input selector for the spectrum analyzer. For each switch position a local oscillator frequency may be defined to translate the center / marker frequency display. At this field, enter the message ID of the switch to monitor. Example: 'S05.position'. Leave this field empty if you don't want to use the frequency translation feature.
inputNames	At this field enter the list of possible selections for the switch that the driver monitors. The switch settings must appear as a comma separated list with no additional space characters. The position names exactly must match the particular settings of the switch device. Example: '11X,11Y,12X,12Y'.
inputLoFreqs	At this field enter the local oscillator frequencies for each switch position, appearing in the same order as the corresponding position names above. Example '10000,10000,11300,11300'.

parameter	*description*
title	If you enter a text into this field, the text is displayed as a title in the spectrum view. You may use this to label the spectrum view.

The spectrum view

The spectrum view shows the spectrum analyzer's output in a similar format as the local display would do. The display shows the measured spectrum (white) in front of a graticule and a summary of instrument settings.



Clicking into the spectrum display activates the marker. You may drag the mouse with the button held down to see a reticule which follows the spectrum curve. This helps to place the marker manually to peaks. More about marker measurements you find at the next paragraph below.

At the upper left corner of the spectrum view you find three 'buttons' which activate additional function when you click to one of them.

- **PRE** --- Launches a separate window showing the instrument memory page
- **PRT** --- Opens the printing dialog, lets you print out the actual spectrum display.
- **SET** --- Launches a separate window showing the instrument settings page.

Marker handling

The spectrum view supports two markers. The main marker is that one provided by the instrument itself. The spectrum view colors this marker yellow. You may set the marker manually by clicking with the left mouse button into the spectrum display. While you hold down the mouse button, a reticule is shown which helps you to place the marker precisely. The instrument settings page provides some common marker functions like 'peak search' or 'marker to center'. These function operate on this main marker.

Additionally the spectrum display provides a delta marker function. To use this function, first place the marker to the location you want to become the reference for the delta measurement. Then set the marker to a second location using the *right* mouse button. This stores the previous marker location as reference (now shown green) to the actual marker location. You may do several delta measurement with the same reference point placing the marker with the right mouse button. Also the peak search and next peak functions work as you might expect. As soon as you place a marker with the left mouse button, the reference point and the delta display gets deleted.

As the marker is based on the instrument's internal marker function, the response to your mouse clicks may be somewhat delayed, specially with a slow communication line. Using the single sweep mode when setting markers improves the performance significantly.

Sweep handling

Viewing the output of a spectrum analyzer at a remote place requires a quite large amount of data to be transferred. Depending on system limitations like the capacity of communication lines this restricts the display update rate to value much lower than the rates you can achieve locally at the instrument.

The spectrum analyzer device driver always operates the instrument in 'single sweep' mode. It manually (i.e. with a remote command) triggers a sweep, when the sweep has finished the trace data is read out and sent to the user interface. The continuous sweep mode selectable at the M&C user interface is implemented in the device driver by repeatedly performing the sequence described above.

Unlike the instrument's local display, the spectrum view shows complete sweeps only. With long sweep times you will find the display frozen until the sweep has completed ans the trace data has been transferred to your user interface. With short sweep times, the maximum sweep rate will be limited by the maximum rate the M&C system is capable to handle and to transfer to the user interface rather than by the sweep time itself.

When you change instrument parameters at the M&C user interface, the device driver

automatically triggers a new sweep if the driver is in single sweep mode. With a slow VLC and/or a slow communication line you will see a noticeable delay between the parameter change an the screen update.

The spectrum analyzers block all communication while they take a measurement (sweep). With smaller RBW settings this may take some seconds. For this reason the communication timeout while reading a measurement is 30 seconds while other commands already time out after 2 seconds.

Display lines

The spectrum display provides four display lines, two horitontal and two vertical lines. The display lines are named F1,F2 and L1,L2 where 'F' stands for frequency and 'L' for level. The display lines are purely implementend in the spectrum display widget, they do not appear on the spectrum analyzer's local display.

The display lines may individually switched on/off, the display line value may either be entered numerically or the lines may be dragged by the mouse in the spectrum display:

At the top of display area, there are four small buttons labelled F1/F2/L1/L2 which control the appearance of the four display lines.

- Clicking to one of the buttons with the left mouse button enables this display line. If previously disabled, the display line is now 'selected', the line and the button are yellow. Clicking again with the left mouse button toggles between the selected and unselected (orange) state.
- Clicking to a button with the right mouse button disables this display line, switched it off.
- The actually selected display line may be dragged do any position within the diagram area.
- While a display line is selected, the normal marker operation is suspended. The marker is shown dimmed and cannot be set with the mouse. To acces the marker again, click once to the display line button which actually appears yellow.

In the parameter panel associated with the spectrum display, the values of the display lines may be set numerically, each display line may be switched on or off individually.

When embedded into user screens at a small size (below 450 pixel width), the spectrum display shows no buttons for display lines as they would interfere with other status information in the display. In such a case either set the display lines numerically or use the standard pop up spectrum display to set the display lines with the mouse. display line parameter are synchronized across multiple display widgets for the same spectrum analyzer, so you can use another, lager instance to set the display lines with the mouse.

Min/Max hold and averaging

The spectrum display window also provides a MIN-HOLD/MAX-HOLD/AVERAGING function. This function is controlled by the 'display mode' parameter at the settings page (or window) for the spectrum analyzer. The following settings are supported:

- NONE --- The display window only shows the each actual sweep (white).

- MAX-HOLD --- The display shows each actual sweep (white) and the maximum values which were found since this mode was enabled (red).
- MIN-HOLD --- The display shows each actual sweep (white) and the minimum values which were found since this mode was enabled (cyan).
- MIN/MAX-HOLD --- The display shows each actual sweep (white) and both, the minimum (cyan) and the maximum (red) values.
- AVG --- The display shows each actual sweep (white) and the averaged values of all sweeps since this mode was enabled. The annotation in the display marking the AVG mode is followed by the number of swees already averaged.

Preset handling

The Rohde & Schwarz FSV driver provides save and load presets function. The Presets are save directly on the driver.

How to configure the driver

The Rohde & Schwarz FSV are controlled via a TCP connection at port 5025. You will configure the spectrum analyzer as the only device at an interface set to the device's IP address and port 5025. To achieve fast display updates, the IDLE time of the interface should be set to a value 100-500 msecs, depening on the communication speed to the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.screen	TEXT	R/O	StringRange R/O
info.options	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.lofreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.memories	TEXT	R/O	StringRange R/O
info.sweepsPerSec	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temperature	FLOAT	R/O	DoubleRange R/O (0 .. 0)
config.inputSelId	TEXT	SAVE SETUP	StringRange
config.inputNames	TEXT	SAVE SETUP	StringRange
config.inputLoFreqs	TEXT	SAVE SETUP	StringRange
config.title	TEXT	SAVE SETUP	StringRange
internal.tick	TEXT	R/O	StringRange R/O
internal.name1	TEXT	R/O	StringRange R/O
internal.name2	TEXT	R/O	StringRange R/O
internal.name3	TEXT	R/O	StringRange R/O
internal.name4	TEXT	R/O	StringRange R/O
internal.name5	TEXT	R/O	StringRange R/O
internal.name6	TEXT	R/O	StringRange R/O
internal.name7	TEXT	R/O	StringRange R/O
internal.name8	TEXT	R/O	StringRange R/O
internal.choices	TEXT	R/O	StringRange R/O
internal.initMessage	TEXT	R/O	StringRange R/O
internal.xpoints	INTEGER	R/O	IntegerRange R/O (0 .. 0)
title	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange ()
byname	CHOICE		EnumRange ()

name	*type*	*flags*	*range*
center	FLOAT		DoubleRange (0.000000 .. 0.000000)
span	FLOAT		DoubleRange (0.000000 .. 0.000000)
span.full	TEXT	NOPRESET	StringRange
span.dn	TEXT	NOPRESET	StringRange
span.up	TEXT	NOPRESET	StringRange
start	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
stop	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
reflev	FLOAT		DoubleRange (0.00 .. 0.00)
reflev.up	TEXT	NOPRESET	StringRange
reflev.dn	TEXT	NOPRESET	StringRange
scale	FLOAT		DoubleRange (0 .. 0)
scale.up	TEXT	NOPRESET	StringRange
scale.dn	TEXT	NOPRESET	StringRange
atten	FLOAT		DoubleRange (0 .. 65)
atten.auto	CHOICE		EnumRange (AUTO FIX)
preamp	CHOICE		EnumRange (OFF ON)
rbw	FLOAT		DoubleRange (0.000 .. 0.000)
rbw.up	TEXT	NOPRESET	StringRange
rbw.dn	TEXT	NOPRESET	StringRange
vbw	FLOAT		DoubleRange (0.000 .. 0.000)
vbw.up	TEXT	NOPRESET	StringRange
vbw.dn	TEXT	NOPRESET	StringRange
sweep	FLOAT		DoubleRange (0.001 .. 4000.000)
rbw.auto	CHOICE		EnumRange (AUTO FIX)
vbw.auto	CHOICE		EnumRange (AUTO FIX)
sweep.auto	CHOICE		EnumRange (AUTO FIX)
trigger	CHOICE	SAVE	EnumRange (CONT SINGLE)

name	*type*	*flags*	*range*
trace	GENERIC	R/O	ObjectRange R/O generic object
sweeping	TEXT	R/O	StringRange R/O
display	CHOICE		EnumRange (NORM MAX HOLD MIN HOLD MIN/MAX HOLD AVG)
marker.x	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
marker.y	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
marker.xpos	INTEGER		IntegerRange (0 .. 0)
marker.mode	CHOICE		EnumRange (OFF ON)
marker.peak	TEXT	NOPRESET	StringRange
marker.nextPeak	TEXT	NOPRESET	StringRange
marker.toCenter	TEXT	NOPRESET	StringRange
marker.toReflev	TEXT	NOPRESET	StringRange
lband.marker.x	FLOAT	R/O	DoubleRange R/O (0.00000000 .. 0.00000000)
foffs	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
foffs.en	CHOICE	SAVE	EnumRange (OFF ON)
loffs	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
loffs.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x1.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.x2.val	FLOAT	SAVE	DoubleRange (0.000000 .. 0.000000)
dl.x2.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y1.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y1.en	CHOICE	SAVE	EnumRange (OFF ON)
dl.y2.val	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
dl.y2.en	CHOICE	SAVE	EnumRange (OFF ON)

name	*type*	*flags*	*range*
reset	TEXT	NOPRESET	StringRange
store	TEXT	NOPRESET	StringRange
recall	TEXT	NOPRESET	StringRange
record	TEXT	NOPRESET	StringRange
state.recording	TEXT	R/O	StringRange R/O
memname.0	TEXT	SAVE	StringRange DIS
memname.1	TEXT	SAVE	StringRange DIS
memname.2	TEXT	SAVE	StringRange DIS
memname.3	TEXT	SAVE	StringRange DIS
memname.4	TEXT	SAVE	StringRange DIS
memname.5	TEXT	SAVE	StringRange DIS
memname.6	TEXT	SAVE	StringRange DIS
memname.7	TEXT	SAVE	StringRange DIS
memname.8	TEXT	SAVE	StringRange DIS
memname.9	TEXT	SAVE	StringRange DIS
memname.10	TEXT	SAVE	StringRange DIS
memname.11	TEXT	SAVE	StringRange DIS
memname.12	TEXT	SAVE	StringRange DIS
memname.13	TEXT	SAVE	StringRange DIS
memname.14	TEXT	SAVE	StringRange DIS
memname.15	TEXT	SAVE	StringRange DIS
internal.fmin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fmax	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.amin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.amax	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.astep	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
lband.center	FLOAT	R/O	DoubleRange R/O (0.0000000000 .. 0.0000000000)
internal.hasB22	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasB24	BOOLEAN	R/O	BooleanRange R/O (true false)

9.584 RuS-PKU-SSPA

This page describes the *Rohde&Schwarz PKU Series SSPA* device driver and its device window.

The drivers was tested with the following equipment:

- Model: PKU-100
- SW version: 2.2.0

To allow proper operation the correct band und model has to be selected in the device setup

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters.
- --- Displays meter readings like output power and BUC specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.

    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- band --- Selects the units band:
    - KU (13,75-14,5GHz),
    - DBS (17.2-18,1GHz) or
    - Lower KU Band (12.75-13.25).
- model --- Selects units power class: 400W or 750W.
- startup --- Selects startup behaviour of the SSPA after a power cycle.
- reference --- Selects reference source of the SSPA
- freqCorrection --- Defines the frequency correction factor in %.
- tempWarn --- Defines the temperature warning threshold level.
- tempFlt --- Defines the temperature fault threshold level.
- linearizer --- En- or Disables the units linearizer.
- linearizerMode --- Selects the linearizer band mode.
- predistortion --- En- or Disables the units predistortion.
- powerLimit --- Defines the units power limit (ALC mode).
- powerWarn --- Defines the units power warning limit in W.
- powerFault --- Defines the units power fault limit in W.
- gainLimit --- Defines the units gain limit (Manual Mode)
- gainWarn --- Defines the units gain warning limit in dB.
- gainFault --- Defines the units gain fault limit in dB.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.hwModel	TEXT	R/O	StringRange R/O
info.hwName	TEXT	R/O	StringRange R/O
info.hwSerial	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.swModel	TEXT	R/O	StringRange R/O
info.acbModel	TEXT	R/O	StringRange R/O
info.acbSerial	TEXT	R/O	StringRange R/O
info.sdbModel	TEXT	R/O	StringRange R/O
info.sdbSerial	TEXT	R/O	StringRange R/O
info.upcModel	TEXT	R/O	StringRange R/O
info.upcSerial	TEXT	R/O	StringRange R/O
config.band	CHOICE	SAVE SETUP	EnumRange (LOW-KU KU DBS)
config.model	CHOICE	SAVE SETUP	EnumRange (400W 750W)
config.startup	CHOICE	SETUP	EnumRange (LAST STATE OFF)
config.reference	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL INT WITH EXTSYNC)
config.freqCorrection	FLOAT	SETUP	DoubleRange (-100.00 .. 100.00)
config.tempWarn	INTEGER	SETUP	IntegerRange (-40 .. 100)
config.tempFlt	INTEGER	SETUP	IntegerRange (-40 .. 100)

name	*type*	*flags*	*range*
config.linearizer	CHOICE	SETUP	EnumRange (ON OFF)
config.linearizerMode	CHOICE	SETUP	EnumRange (NARROW-BAND HIGH-1)
config.powerLimit	INTEGER	SETUP	IntegerRange (0 .. 202)
config.powerWarn	INTEGER	SETUP	IntegerRange (0 .. 202)
config.powerFault	INTEGER	SETUP	IntegerRange (0 .. 202)
config.gainLimit	FLOAT	SETUP	DoubleRange (0.5 .. 3.0)
config.gainWarn	FLOAT	SETUP	DoubleRange (-10.0 .. 0.0)
config.gainFault	FLOAT	SETUP	DoubleRange (-10.0 .. 0.0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
internal.name	TEXT	R/O	StringRange R/O
internal.model1	TEXT	R/O	StringRange R/O
internal.model2	TEXT	R/O	StringRange R/O
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.frequency	INTEGER		IntegerRange (0 .. 0)
rx.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.attn	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.fan	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.efficiency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.acPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.on	CHOICE		EnumRange (ON OFF)
tx.gain	FLOAT		DoubleRange (0.0 .. 200.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 200.0)

name	*type*	*flags*	*range*
tx.controlMode	CHOICE		EnumRange (NONE ALC)
tx.requestedLevel	FLOAT		DoubleRange (-43.0 .. 23.1)
rx.requestedLevel	INTEGER		IntegerRange (-25 .. 0)
rx.attn	INTEGER		IntegerRange (0 .. 25)
rx.gain	INTEGER		IntegerRange (0 .. 25)
rx.measuredLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Driver)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Overdrive)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Gate temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Initialization)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal communication)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Invalid configuration)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Drain current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Gate current)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PA transistor)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Power limiter)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Ext shutdown)
faults.13	ALARM	R/O	AlarmFlagRange R/O (System start)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Transistor)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Gate supply Warning)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Gate supply)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Ext. reference)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PLL 100 unlocked)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (PLL LO unlocked)
faults.20	ALARM	R/O	AlarmFlagRange R/O (OCXO)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input Warning)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Input)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input low)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Input high)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Output muted)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Output reduced)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Reflected pwr)
faults.29	ALARM	R/O	AlarmFlagRange R/O (RF Power Warning)
faults.30	ALARM	R/O	AlarmFlagRange R/O (RF Power)
faults.31	ALARM	R/O	AlarmFlagRange R/O (CG power limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Regulation level)
faults.33	ALARM	R/O	AlarmFlagRange R/O (DC Warning)
faults.34	ALARM	R/O	AlarmFlagRange R/O (DC)
faults.35	ALARM	R/O	AlarmFlagRange R/O (DC1)
faults.36	ALARM	R/O	AlarmFlagRange R/O (DC2)
faults.37	ALARM	R/O	AlarmFlagRange R/O (SDB board supply)
faults.38	ALARM	R/O	AlarmFlagRange R/O (PSU)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Temperature Warn)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU1)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (Temperature PSU2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (NTP)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Remote)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.585 S3Satcom-ACU

This page describes the device driver and the device window for the S3SatCom-ACU via serial communication protocol Tadiran-Scopus.

The default TCP/IP communication port of the ACU is 2008

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the tracking state.
- --- Contains the tracking parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*
address	There is no address required.
azLoLimit	Sets the lower azimuth travel limit.
azHiLimit	Sets the upper azimuth travel limit.
elLoLimit	Sets the lower elevation travel limit.
elHiLimit	Sets the upper elevation travel limit.

parameter	*description*
plLoLimit	Sets the lower polarization travel limit.
plHiLimit	Sets the upper polarization travel limit.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.polarization	TEXT	R/O	StringRange R/O
info.beacon.1	TEXT	R/O	StringRange R/O
info.beacon.2	TEXT	R/O	StringRange R/O
info.iduVersion	TEXT	R/O	StringRange R/O
info.oduVersion	TEXT	R/O	StringRange R/O
info.motor	TEXT	R/O	StringRange R/O
info.heater	TEXT	R/O	StringRange R/O
info.diameter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.classic	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state11	HEX	R/O	HexRange R/O (0 .. 0)
internal.state12	HEX	R/O	HexRange R/O (0 .. 0)
internal.state13	HEX	R/O	HexRange R/O (0 .. 0)
internal.state14	HEX	R/O	HexRange R/O (0 .. 0)
internal.state2	HEX	R/O	HexRange R/O (0 .. 0)
internal.bulk2	HEX	R/O	HexRange R/O (0 .. 0)
internal.bulk8	HEX	R/O	HexRange R/O (0 .. 0)
internal.limits	TEXT	R/O	StringRange R/O
internal.track1	TEXT	R/O	StringRange R/O
internal.track2	TEXT	R/O	StringRange R/O
internal.powerfail	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.polset	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.aztrack	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.poltrack	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hemisphere	BOOLEAN	R/O	BooleanRange R/O (true false)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.000 .. 360.000)

name	*type*	*flags*	*range*
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.000 .. 360.000)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.000 .. 90.000)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.000 .. 90.000)
config.plLoLimit	FLOAT	SETUP	DoubleRange (-180.000 .. 180.000)
config.plHiLimit	FLOAT	SETUP	DoubleRange (-180.000 .. 180.000)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polPlane	CHOICE		EnumRange (X Y)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.beacon.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.beacon.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
trackMode	CHOICE		EnumRange (STANDBY STEP NORAD)
st.interval	INTEGER		IntegerRange (1 .. 60)
no.interval	INTEGER		IntegerRange (1 .. 60)
measDelay	INTEGER	R/O	IntegerRange R/O (1 .. 60)
loss	FLOAT	R/O	DoubleRange R/O (0.1 .. 3.0)
meas.iduTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.oduTemp1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.oduTemp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.cpuTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.motorTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.azVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.elVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.plVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
longitude	FLOAT		DoubleRange (-180.000000 .. 180.000000)
latitude	FLOAT		DoubleRange (-90.000000 .. 90.000000)
altitude	INTEGER		IntegerRange (0 .. 0)
motorStop	TEXT	NOPRESET	StringRange
emergencyStop	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
state.psuAz	BOOLEAN	R/O	BooleanRange R/O (true false)
state.psuEl	BOOLEAN	R/O	BooleanRange R/O (true false)
state.psuPl	BOOLEAN	R/O	BooleanRange R/O (true false)
heater	CHOICE	R/O	EnumRange R/O (OFF HEATING COOLING)
mode.azTrack	CHOICE	R/O	EnumRange R/O (ENABLED DISABLED)
mode.plTrack	CHOICE	R/O	EnumRange R/O (ENABLED DISABLED)
powerfail	CHOICE	R/O	EnumRange R/O (STANDBY RESTORE)
polset	CHOICE	R/O	EnumRange R/O (NO SET THEORETICAL)
hemisphere	CHOICE	R/O	EnumRange R/O (NORTH SOUTH)
target.name	TEXT	R/O	StringRange R/O
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Emergency stop FP)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Emergency stop Ext/ODU)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ encoder)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL encoder)
faults.10	ALARM	R/O	AlarmFlagRange R/O (PL encoder)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AZ VFD)
faults.12	ALARM	R/O	AlarmFlagRange R/O (EL VFD)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Heater/Cooler)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ODU Comms)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.16	ALARM	R/O	AlarmFlagRange R/O (AZ PSU)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL PSU)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PL PSU)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Beacon RX1 Summary)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Beacon RX1 Tracking)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Beacon RX1 Summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Beacon RX1 Tracking)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Tracking Processor)

9.586 SCC-Device

The SCC-Device logical device manages the equipment which is necessary to let the NMS control a VLC via SCC (Satellite Control Channel). In each SCC controlled VLC there must be exactly one SCC-Device, this device **must** be named **SCC** .

In the simples case, the SCC-Device controls the satellite modem which implements the physical data link to the NMS. The SCC-Device als is prepared to manage more complex equipment setups where additional equipment like frequency converters or power amplifiers are involved in the reception or transmission of the SCC signals.

Even though not mandatory, it makes sense to give the modem used for SCC the same name on every VLC. E.g. the recovery macro described at the bottom of this chapter may be identical for all VLCs if the devices are named the same.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual control channel state. When viewed from the NMS, one always will see 'receiving data from the NMS' marked green. Using a client which is directly connected to the VLC via Ethernet, the state of the connection to the NMS can be monitored by means of this display.
- --- The faults page. The SCC-Device only knows a 'communication fault' which signals that the heartbeat messages from the NMS are missing.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the SCC-Device to the devices it monitors / controls. Unused parameters should be left empty.

- *parameter --- description*
- txOnId --- The parameter ID used to switch the SCC carrier on/off which VLC transmits during a point to point connection. Usually something like 'SCC-MODEM.tx.on'.
- txPwrId --- The parameter ID of the setting which controls the transmit power while the VLC transmits an SCC signal. Example: 'SCC-MODEM.tx.power'.
- txPwrCal --- In the NMS, the definition of the SCC inbound channel the VLC uses contains the transmit power expressed as EIRP (dBW). As the 'txPwrId' usually sets a modem output level, the 'txPwrCal' is used to convert these values:
  - If the SCC-Device directly controls a modem output level, set the 'txPwrCal' value to the EIRP (dBW), the VLC (theoretically) transmits with the modem output level set to 0 dBm.
  - If the SCC-DEvice is configured to set the transmit power through a 'Level-Control'

or a 'Level-Set' logical device, set 'txPwrCal' to zero and configure the EIRP calibration in the device controlling the power.

- txDataRateId --- The parameter ID of the setting which controls the transmit data rate of the inbound data stream. Example: 'SCC-MODEM.tx.mod.dataRate'.
- txFrequencyId --- The parameter ID of the setting which controls the transmit frequency. Example: 'SCC-MODEM.tx.frequency'.
- txFrequencyOffs --- The 'txFrequencyOffs' value gets subtracted from the SCC channel transmit frequency to achieve the frequency set at the modem. If you set the frequency through a 'Frequency-Split' logical device rather than controlling the modem directly, set this value to zero and let the Frequency-Split device manage the frequency offsets.
- rxDataRateId --- The parameter ID of the setting which controls the receive data rate of the outbound data stream. Example: 'SCC-MODEM.rx.dem.dataRate'.
- rxFrequencyId --- The parameter ID of the setting which controls the receive frequency. Example: 'SCC-MODEM.rx.frequency'.
- rxFrequencyOffs --- The 'rxFrequencyOffs' value gets subtracted from the SCC channel receive frequency to achieve the frequency set at the modem. If you set the frequency through a 'Frequency-Split' logical device rather than controlling the modem directly, set this value to zero and let the Frequency-Split device manage the frequency offsets.
- timeout --- If the SCC-Device does not recognize the NMS' heartbeat for the time specified here, it assumes that the channel is down and starts to scan the two outbound channel definitions it received from the NMS.

Recovery through macro

If the SCC-Device misses the heartbeat messages from the NMS, it scans the two OUTBOUND channel definitions it did receive from the NMS before. Before setting the frequency and data rate defined for a channel, the SCC-Device checks if there is a macro called 'scc-recovery' in the 'macros' subdirectory. If such a macro is present it gets executed before the channel frequency and data rate is set.

Such a recovery macro should contain all device settings necessary to receive the NMS' outbound channel. Beside general modem settings this macro may include the position of waveguide switches, the frequency setting of a downconverter or other parameters. It however must not contain a command which switches on the SCC carrier in the VLC.

If a parameter the VLC requires to receive the NMS outbound channel gets displaced by accident, the communication with the NMS immediately breaks down. This mistake cannot be undone via SCC as the VLC does not receive any data from the NMS. If this parameter is backed up in the scc-recovery macro, the SCC-Device successfully will reinitialize the communication equipment during the recovery / channel scanning process.

A recovery macro gets recorded like any other macro while the VLC is 'online'. TheMacro Management window provides a particular toolbar button which lets you store a macro as a scc-recovery macro on the VLC with a single mouse click.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.broadcastIp	TEXT	SAVE SETUP	StringRange
config.txOnId	TEXT	SAVE SETUP	StringRange
config.txPwrId	TEXT	SAVE SETUP	StringRange
config.txPwrCal	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.txDataRateId	TEXT	SAVE SETUP	StringRange
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.txFrequencyOffs	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.rxDataRateId	TEXT	SAVE SETUP	StringRange
config.rxFrequencyId	TEXT	SAVE SETUP	StringRange
config.rxFrequencyOffs	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.timeout	INTEGER	SAVE SETUP	IntegerRange (10 .. 600)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
command	TEXT		StringRange
receiveOK	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
search1	BOOLEAN	R/O	BooleanRange R/O (true false)
search2	BOOLEAN	R/O	BooleanRange R/O (true false)

9.587 SES-DAKS

This page describes the SES Daks device driver. It sends a message to the SES Daks system via N2P proxy server. Therefore it requires an helper script (Perl) in /home/satnms/ses-daks.pl. The message could be triggered by any event.

The message is sent if one of two configurable parameters in the system changes to the reference value defined in the device configuration.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the N2P proxy server as well as defining the notification event.

parameter	*description*
input.Id	Defines the parameter triggering an event (e.g. SUMMARY.fault).
input.compare	Defines the compare condition to Equal or Not Equal .
input.reference	Defines the condition value of the parameter.
input.Id2	Defines a second parameter triggering an event (e.g. SUMMARY.fault).
input.compare2	Defines the compare condition to Equal or Not Equal .
input.reference2	Defines the condition value of the second parameter.
ipAddr	Defines the ip address of the N2P .
tcpPort	Defines the tcp port of the N2P .

parameter	*description*
bcid	Defines the bcid of the N2P.
annid	Defines the annid of the N2P proxy.
msgtxt	Defines the message text send to the N2P .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.lastReturn	TEXT	R/O	StringRange R/O
info.lastMessage	TEXT	R/O	StringRange R/O
info.returnCode	TEXT	R/O	StringRange R/O
config.input.Id	TEXT	SAVE SETUP	StringRange
config.input.compare	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input.reference	TEXT	SAVE SETUP	StringRange
config.input.Id2	TEXT	SAVE SETUP	StringRange
config.input.compare2	CHOICE	SAVE SETUP	EnumRange (EQ NE)
config.input.reference2	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ipAddr	TEXT	SAVE SETUP	StringRange
config.tcpPort	TEXT	SAVE SETUP	StringRange
config.bcid	TEXT	SAVE SETUP	StringRange
config.annid	TEXT	SAVE SETUP	StringRange
config.msgtxt	TEXT	SAVE SETUP	StringRange
input	TEXT		StringRange
input2	TEXT		StringRange

9.588 SES-Sysinfo

The SES-Sysinfo device displays parameters of the server machine. This includes disk usage, cpu load, memory and sensor information from the mainboard.

Unlike the Sysinfo device it is intended for monitoring of operating system parameters for IBM/Lenovo system x3250 M5 and x3250 M6 server.

It supports the following sat-nms MNC types:

- 3.70 - 3.99 hardware IBM x3250 M5 with debian wheezy 7.6+ => internal.os = 3
- 3.70 - 3.99 hardware IBM x3250 M6 with debian wheezy 7.11 => internal.os = 3

It is possible to configure alarm thresholds to create alarm events if the disk capacity goes low, if the 15min average cpu load is to high, if the available memory is to low or if a temperature is to high.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows cpu load and disk usage etc.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described

below.

Configuration parameters

parameter	*description*
diskThreshold	The threshold for the available disk capacity
hasHotspare	Defines if a hotspare RAID drive is available
loadThreshold	The threshold for the 15min average cpu load
memThreshold	The threshold for RAM
pchThreshold	The threshold for PCH temperature.
pciThreshold	The threshold for PCI temperature.
vrdThreshold	The threshold for VRD temperature.
cpuThreshold	The threshold for CPU temperature.

Fault flags

The device provides the following fault flags:

- 'low disk capacity': the available space on the internal disk is lower than the configured threshold
- 'high average load': the average cpu load of the last 15min was higher than the configured threshold
- 'low memory': the available memory (RAM) is lower than the configured threshold
- 'high .. temperature': the temperature is higher than the configured threshold
- 'RAID volume': one of the raid volume failed
- 'RAID (hotspare) disk ..': the respective raid disk is not in optimal condition

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.raidController	TEXT	R/O	StringRange R/O
info.hostname	TEXT	R/O	StringRange R/O
info.cores	TEXT	R/O	StringRange R/O
config.diskThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hasHotspare	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.loadThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.pchTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.pciTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.vrdTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.cpuTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
disk.name	TEXT	R/O	StringRange R/O
disk.size	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.available	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
disk.usage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
raid.enabled	TEXT	R/O	StringRange R/O
raid.operation	TEXT	R/O	StringRange R/O
raid.state	TEXT	R/O	StringRange R/O
raid.rebuild	TEXT	R/O	StringRange R/O
raid.disk1.state	TEXT	R/O	StringRange R/O
raid.disk1.model	TEXT	R/O	StringRange R/O
raid.disk1.vendor	TEXT	R/O	StringRange R/O
raid.disk1.serial	TEXT	R/O	StringRange R/O
raid.disk2.state	TEXT	R/O	StringRange R/O
raid.disk2.model	TEXT	R/O	StringRange R/O
raid.disk2.vendor	TEXT	R/O	StringRange R/O
raid.disk2.serial	TEXT	R/O	StringRange R/O
raid.disk3.state	TEXT	R/O	StringRange R/O DIS
raid.disk3.model	TEXT	R/O	StringRange R/O DIS
raid.disk3.vendor	TEXT	R/O	StringRange R/O DIS
raid.disk3.serial	TEXT	R/O	StringRange R/O DIS
mem.total	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.free	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.uptime	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
stat.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature.pch	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.pci	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
temperature.vrd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.ambient	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.cpu	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.speed.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mail.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mail.subjects	TEXT	R/O	StringRange R/O
internal.version	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.os	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (low disk capacity)
faults.02	ALARM	R/O	AlarmFlagRange R/O (high average load)
faults.03	ALARM	R/O	AlarmFlagRange R/O (high pch temp)
faults.04	ALARM	R/O	AlarmFlagRange R/O (high pci temp)
faults.05	ALARM	R/O	AlarmFlagRange R/O (high vrd temp)
faults.06	ALARM	R/O	AlarmFlagRange R/O (high cpu temp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (low memory)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (RAID volume)
faults.11	ALARM	R/O	AlarmFlagRange R/O (RAID disk 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (RAID disk 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (RAID hotspare disk)
faults.14	ALARM	R/O	AlarmFlagRange R/O (new mail)
faults.20	ALARM	R/O	AlarmFlagRange R/O (unknown RAID Ctrl)

9.589 SNG-Pointing

The SNG-Pointing logical device is a 'high level' driver for motor driven, car-mounted antennas. It is used together with a driver for the type of antenna controller used. In order to work properly together with the SGN-Pointing device, the antenna controller / driver must provide parameters to read and command the antenna elevation in terms of 'world coordinates'. The SNG-Pointing device does not make any calculations to include the nick/roll angles of the car into the antenna pointing angles.

The features of the SNG-Pointing device are:

- It supports the operator in calibrating the azimuth angle for the actual position of the car. The azimuth calibration is done in the traditional way: Setting the elevation to the value of a known satellite and then scanning the azimuth until the satellite signal is received. Once the antenna pointing is optimized for this satellite, pressing the 'Calibrate' button adds an internal offset to the azimuth, making it read the value calculated from the satellite's orbit position and the geodetic location of the antenna.
- It permits to set the antenna pointing by supplying a satellite orbit position. The Antenna-Pointing logical device computes the antenna pointing from the satellite position and sends these values to the low level antenna driver. The device may read the geodetic location automatically from a GPS receiver (this must be present as a NMEA-GPS-Receiver device) or the operator may enter the location parameters manually.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing values, the selected satellite name and the orbit position of this satellite. You should use this page if you want to move the antenna to a certain orbit position. For details, read the paragraph 'Operation' below.

- --- This page contains the antenna location coordinates. The values may be entered manually if there is no GPS receiver configured or if GPS positioning is not possible at the actual location. As soon as the GPS receiver has determined the geodetic position, any manually entered values get over written.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

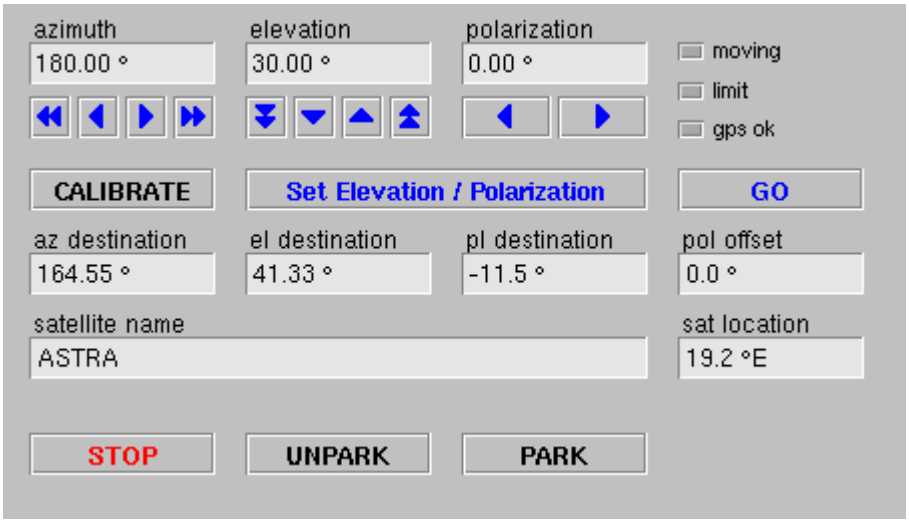
Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the SNG-Pointing device work properly.

parameter	***description***
acuDevice	The device name of the antenna controller.
antennaParkId	The message ID used to park the antenna.
antennaParkCmd	The value / command sent to the ID above.
antennaUnparkId	The message ID used to unpark the antenna.
antennaUnparkCmd	The value / command sent to the ID above.
gpsDevice	The device name of the GPS receiver. Leave this field empty if there is no GPS receiver available.
azStep	The azimuth displacement applied when the operator clicks a (single arrow) jog button. The large step buttons (double arrow) move the antenna ten times of this value.
elStep	The elevation displacement applied when the operator clicks a (single arrow) jog button. The large step buttons (double arrow) move the antenna ten times of this value.
plStep	The polarization displacement applied when the operator clicks a jog button.
azOffset	The azimuth calibration offset. The value usually gets determined with the calibration procedure described below, there is no need to set the offset manually.

Operation

The primary page of the device window is used to operate the antenna. The first row of parameters show the actual antenna pointing. The azimuth reading already implies the offset derived from the last calibration procedure.



The blue arrow buttons allow to move the antenna in small or larger steps. The step size is configurable individually for each axis at the device setup page. Repeatedly pressing a button accumulates the displacements.

The **CALIBRATE** button recalculates the azimuth offset to make the actual azimuth value read the value in the 'az destination' field. The calculated offset is stored permanently, even if the VLC is switched off.

The **GO** button commands the antenna to move to the pointing angles shown in the 'destination' fields. **Set Elevation/Polarization** does the same, but does not move the azimuth. The latter button is intended to prepare the antenna for the azimuth calibration procedure.

The 'destination' pointing angles are either manually supplied or values calculated from the satellite's orbit position. Changing these values does *not* move the antenna until **GO** or **Set Elevation/Polarization** is pressen.

'pol offset' is a satellite-depending parameter describing the offset to be added to the theoretically calculated polarization angle in order to optimize the polarization separation for this satellite. If you change this value, the 'destination' angles get recalculated from the new polarization offset and the satellite location.

Changing the satellite location parameter ('sat location') recalculates the destination pointing angles, taking the actual 'pol offset' into account. The antenna is not moved by this.

Preset Storage

The parameters of frequently used satellites may be stored as device presets of the SNG-Pointing device. The satellite name, orbit position and polarization offset are the only values stored in the preset. Recalling the preset means updating these values. In succession the device also recalculates the destination pointing angles to the antenna pointing valid for the actual antenna location. Finally **GO** must be pressed to move the antenna to the selected satellite.

Calibration Process

The SNG-Pointing device supports the operator in calibrating the azimuth offset for the actual antenna location / setup. This usually is done in the following way:

1. Ensure, that the SNG-Pointing device knows the precise geodetic location of the antenna. This may happen automatically with the values supplied by the GPS receiver or the location coordinated must be entered manually.
2. Unpark the antenna. This moves up the antenna, leaving the azimuth in park position.
3. The azimuth reading still includes the calibration offset from the previous application of the antenna. If you find the obviously wrong reading bothersome, reset the offset at the device's setup page or do a first calibration by eye: Enter the azimuth value you are guessing the antenna is viewing to in the 'az destination' field an press **CALIBRATE** .
4. Now select a known satellite as the reference to calibrate the antenna to. Either select the satellite from the list of stored presets or enter the orbit position of the satellite in the 'sat location' field. The 'destination' fields now read the antenna pointing angles for this satellite.
5. Now press **Set Elevation/Polarization** to set the elevation angle. If you did a coarse calibration by eye before, you may press **GO** , which als moves the azimuth roughly to where the satellite is expected.
6. Watch your receiver or spectrum analyzer and horizontally move the antenna in small steps until you found the satellite.
7. Fine-tune the antenna adjustment for a max. receive level.
8. Press **CALIBRATE** to store the azimuth offset.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.acuDevice	TEXT	SAVE SETUP	StringRange
config.antennaParkId	TEXT	SAVE SETUP	StringRange
config.antennaParkCmd	TEXT	SAVE SETUP	StringRange
config.antennaUnparkId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.antennaUnparkCmd	TEXT	SAVE SETUP	StringRange
config.gpsDevice	TEXT	SAVE SETUP	StringRange
config.azStep	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.elStep	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.plStep	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.azOffset	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.presetDefault	TEXT	R/O	StringRange R/O
azimuth	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
elevation	FLOAT	NOPRESET	DoubleRange (0.00 .. 0.00)
polarization	FLOAT	NOPRESET	DoubleRange (0.0 .. 0.0)
orbit	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
satname	TEXT	SAVE	StringRange
poloffset	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
step	TEXT	NOPRESET	StringRange
park	TEXT	NOPRESET	StringRange
unpark	TEXT	NOPRESET	StringRange
go	TEXT	NOPRESET	StringRange
goelpl	TEXT	NOPRESET	StringRange
calibrate	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.aped	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.gpsOk	BOOLEAN	R/O	BooleanRange R/O (true false)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
longitude	FLOAT	SAVE NOPRESET	DoubleRange (0.0000 .. 0.0000)
latitude	FLOAT	SAVE NOPRESET	DoubleRange (0.0000 .. 0.0000)
altitude	FLOAT	SAVE NOPRESET	DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Antenna Controller)
faults.02	ALARM	R/O	AlarmFlagRange R/O (GPS Receiver)

9.590 SNMP-Driver-Template

The *SNMP-Driver-Template* is not a functional device driver. It is meant as a template which makes it easy to write a small SNMP device driver which simply polls / controls some parameters at the device.

To make use of this template, go into the 'drivers' directory if your sat-nms installation and make a copy of the 'SNMP-Driver-Template.device' file. To match the sat-nms naming conventions, the name of the new file should be Vendor-MyDevice.device or something similar (You may be unable to select your device driver in the device setup window if the file is not named correctly).

Edit the new file, the comment-header in the file contains some instructions how to adapt the driver file for your needs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the defined parameters
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- retries --- The number of communication retries which are made at application level. Should be 0 in most cases as the SNMP protocol already provides a retry-mechanism internally.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
param.1.val	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
param.1.desc	TEXT	R/O	StringRange R/O
param.2.val	INTEGER	R/O	IntegerRange R/O (0 .. 0)
param.2.desc	TEXT	R/O	StringRange R/O
param.3.val	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
param.3.desc	TEXT	R/O	StringRange R/O
param.4.val	CHOICE	R/O	EnumRange R/O (VALUE-A VALUE-B VALUE-C VALUE-D)
param.4.desc	TEXT	R/O	StringRange R/O
param.5.val	BOOLEAN	R/O	BooleanRange R/O (true false)
param.5.desc	TEXT	R/O	StringRange R/O
param.6.val	TEXT		StringRange
param.6.desc	TEXT	R/O	StringRange R/O
param.7.val	INTEGER		IntegerRange (0 .. 0)
param.7.desc	TEXT	R/O	StringRange R/O
param.8.val	FLOAT		DoubleRange (0.00 .. 0.00)
param.8.desc	TEXT	R/O	StringRange R/O
param.9.val	CHOICE		EnumRange (VALUE-A VALUE-B VALUE-C VALUE-D)
param.9.desc	TEXT	R/O	StringRange R/O
param.10.val	BOOLEAN		BooleanRange (true false)
param.10.desc	TEXT	R/O	StringRange R/O

9.591 SPX-Etsiline-CommPact-v101

This page describes the device driver and the device window for SPXFLOW Etsiline CommPact dehydrator.The driver is intended for monitoring only purpose.

For SW releases later than **1.02** use SPX-Etsiline-CommPact device driver

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as to allow proper alarm monitoring.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
- pressureLowAlarm --- Defines the lower limit for the device pressure in mBar which triggers the low pressure fault.
- pressureHighAlarm --- Defines the upper limit for the device pressure in mBar which triggers the high pressure fault.
- humidityAlarm --- Defines the upper limit for the device humidity in percent which triggers the high humidity fault.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.pressureLowLevel	INTEGER	SETUP	IntegerRange (0 .. 0)
config.pressureHighLevel	INTEGER	SETUP	IntegerRange (0 .. 0)
config.pressureLowAlarm	INTEGER	SAVE SETUP	IntegerRange (1 .. 7000)
config.pressureHighAlarm	INTEGER	SAVE SETUP	IntegerRange (1 .. 7000)
config.humidityAlarm	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity	INTEGER	R/O	IntegerRange R/O (0 .. 100)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High humidity)

9.592 SPX-Etsiline-CommPact

This page describes the device driver and the device window for SPXFLOW Etsiline CommPact dehydrator.

For SW releases earlier than **1.02** use SPX-Etsiline-CommPact-1-01 device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device as well as to allow proper alarm monitoring.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized or a proved preset value which should not be changed under normal circumstances.
- pressureLowLevel --- Defines the low pressure operation point .
- pressureHighLevel --- Defines the high pressure operation point.
- pressureLowAlarm --- Defines the lower limit for the device pressure in mBar which triggers the low pressure fault.
- pressureHighAlarm --- Defines the upper limit for the device pressure in mBar which triggers the high pressure fault.

- humidityAlarm --- Defines the upper limit for the device humidity in percent which triggers the high humidity fault. Default humidity alarm is set on 7% RH, which is a dew point of -15°C at 22°C ambient.
- alarmDelay --- Defines the delay time for alarms in seconds. When an alarm restores within de delay time, no alarm signal will be given.
- pressureHysteresis --- Defines the alarm pressure hysteresis for low and high pressure alarms between 5mBar and 250mBar.
- runtimeAlarm --- Defines the compressor runtime alarm threshold in minutes. When the compressor runs more than the set running hours, a run time alarm is given.
- humidityEnable --- Enables monitoring of a humidity sensor if supported by the device .
- dewpointEnable --- Enables monitoring of the dew point if supported by the device.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.runtime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.runtimeComp1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.runtimeComp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.config	CHOICE	R/O	EnumRange R/O (ONE PUMP TWO PUMPS)
info.sensCalibration	INTEGER	R/O	IntegerRange R/O (0 .. 2000)
info.sensRange	INTEGER	R/O	IntegerRange R/O (1 .. 150)
config.pressureLowLevel	INTEGER	SETUP	IntegerRange (1 .. 7000)

name	*type*	*flags*	*range*
config.pressureHighLevel	INTEGER	SETUP	IntegerRange (1 .. 7000)
config.pressureLowAlarm	INTEGER	SETUP	IntegerRange (1 .. 7000)
config.pressureHighAlarm	INTEGER	SETUP	IntegerRange (1 .. 7000)
config.humidityAlarm	INTEGER	SETUP	IntegerRange (0 .. 100)
config.alarmDelay	INTEGER	SETUP	IntegerRange (0 .. 0)
config.pressureHysteresis	INTEGER	SETUP	IntegerRange (5 .. 250)
config.runtimeAlarm	INTEGER	SETUP	IntegerRange (0 .. 60)
config.humidityEnable	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.dewpointEnable	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
internal.version	TEXT	R/O	StringRange R/O
internal.time1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pressure	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity	INTEGER	R/O	IntegerRange R/O (0 .. 100)
dewpoint	FLOAT	R/O	DoubleRange R/O (-60.00 .. 60.00)
state	CHOICE	R/O	EnumRange R/O (COOLING DRYER 1 HEATING DRYER 1 COOLING DRYER 2 HEATING DRYER 2)
runtime.heat1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
runtime.cool1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
runtime.heat2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
runtime.cool2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
dutyCycle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
position.heat1	CHOICE	R/O	EnumRange R/O (OPEN CLOSED)

name	*type*	*flags*	*range*
position.heat2	CHOICE	R/O	EnumRange R/O (OPEN CLOSED)
position.pump1	CHOICE	R/O	EnumRange R/O (OPEN CLOSED)
position.pump2	CHOICE	R/O	EnumRange R/O (OPEN CLOSED)
valveY1	CHOICE	R/O	EnumRange R/O (ON OFF)
valveY2	CHOICE	R/O	EnumRange R/O (ON OFF)
valveY3	CHOICE	R/O	EnumRange R/O (ON OFF)
valveY4	CHOICE	R/O	EnumRange R/O (ON OFF)
testmode	CHOICE	R/O	EnumRange R/O (ACTIVE NOT ACTIVE)
voltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
alarmRelay	CHOICE	R/O	EnumRange R/O (OPEN CLOSED)
sensEnable	CHOICE	R/O	EnumRange R/O (DISABLED ENABLED)
sensConnection	CHOICE	R/O	EnumRange R/O (I2C ANALOG)
sensUnits	CHOICE	R/O	EnumRange R/O (RELATIVE HUMIDITY DEWPOINT)
remSensEnable	CHOICE	R/O	EnumRange R/O (DISABLED ENABLED)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low pressure)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High pressure)
faults.03	ALARM	R/O	AlarmFlagRange R/O (High humidity)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Compressor runtime)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Test state)

9.593 SSE-ASat

This page describes the device driver and the device window for the SSE S-Series transceivers. The device driver is mostly self-configuring, it reads options like the frequency range or the presence of the TX/RX parts in the transceiver from the unit and sets up the list of available M&C parameters accordingly.

The device driver uses the NPI packet protocol to control the transceiver when operated at a RS485 interface. This protocol allows multiple SSE devices to be controlled via the same physical interface.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- RX --- This page shows receive part parameters of the device. If the device is configured to "TX-ONLY" mode, all parameters of this page remain gray.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	***description***
address	The device driver expects the NPI address (LU, logical unit) of the transceiver entered here as a two digit, hexadecimal number. You should use upper case letters for the hex digits A..F, as the transceivers requires the address to be of this format. If the RS232 interface is used together with the Terminal-CR-CRLF-E protocol, the address is ignored, you may leave this field empty in this case.
powerOnDelay	You can force the transceiver to inhibit the transmitted signal for this time after power on. Setting the value to zero disables this feature.
txAutoOn	Defines the default TX ON/OFF state when the unit gets powered on.
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. The transceiver either expects *1200,O,7,1* or *9600,N,8,1* for the RS232 interface, depending on the cable layout. The RS485 line speed can be set with a command at the RS232 port.

Communication protocol / Interface type

This device either may be controlled via a RS422/RS485 interface using the SSE-NPI packet protocol or via a RS232 interface using the text based Terminal-CR-CRLF-E protocol. It is recommended to use the RS485 interface as the NPI protocol, which is only available with this interface type, provides a true multidrop device addressing and a checksum protection for the data packets exchanged.

Regardless of the interface type used, the transceiver must be placed into the NPI protocol mode before it can be used together with the software. Connect a RS232 terminal or a PC running a terminal program to the RS232 port of the transceiver and enter the command

"NPI=ON" to program this setting.

Baud rate select

The RS232 interface of the S-Series transceivers normally works at 1200 baud, 7 bits, odd parity. You may connect the "baud select" line (pin H) to ground (pin C) in the transceiver's communication port connector to switch the line settings to *9600,N,8,1* .

The speed of the RS485 port in NPI mode is programmed by a terminal command which has to be entered at the RS232 port. Hence, to set the RS485 speed, you have to connect a RS232 terminal or a PC running a terminal program to the RS232 port of the transceiver. The RS485 line parameters always are 8 bit, no parity in NPI mode.

RS485 cabling

The SSE manual describes a 2-wire RS485 cabling with special termination circuits at both ends of the line for the NPI mode. The RS422/RS485 interface cards typically used with a VLC or M&C system do not support this cabling! Use the 4-wire cabling described for the "RS 485 multidrop mode" instead. It works fine with the NPI mode, too.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerOnDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 600)
config.txAutoOn	CHOICE	SAVE SETUP	EnumRange (OFF ON)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.rfband	TEXT	R/O	StringRange R/O
info.ifband	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.mode	TEXT	R/O	StringRange R/O
info.maxpower	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.twta	TEXT	R/O	StringRange R/O
info.atten	TEXT	R/O	StringRange R/O
info.synths	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.frequency	FLOAT		DoubleRange (0 .. 12750)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 100.0)
tx.frequency	FLOAT		DoubleRange (0 .. 14500)
tx.gain	FLOAT		DoubleRange (0 .. 31)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.state	HEX	R/O	HexRange R/O (0 .. 0)
internal.tx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.tx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rpm	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RX Level)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FFL1 Lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (FFL1 Level)
faults.04	ALARM	R/O	AlarmFlagRange R/O (FAL1 Lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (FAL2 Lock)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (HFLO1 Level)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HFLO2 Level)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TX Level)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Supply +15V)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Supply +10V)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Supply +5V)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Supply -5V)
faults.13	ALARM	R/O	AlarmFlagRange R/O (FFL2 Lock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (FFL2 Level)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Supply +12V)
faults.16	ALARM	R/O	AlarmFlagRange R/O (1400 MHz Lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (1200 MHz Lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (750 MHz Lock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TWT Summary)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TWT Standby)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWT Beam Off)

9.594 SSE-KStar

This page describes the device driver and the device window for the SSE K-Star transceiver. The device driver is mostly self-configuring, it read options like the frequency range or the presence of the TX/RX parts in the transceiver from the unit and sets up the list of avaliable M&C parameters accordingly.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- RX --- This page shows receive part parameters of the device. If the device is configured to

"TX-ONLY" mode, all parameters of this page remain gray.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	***description***
address	You must set this parameter to the transceiver's device address (1..7) if you intend to control the unit via RS485. If you are using the transceiver's RS232 port for M&C, leave the address field empty, this prevents the driver from sending an initial "SAD" command to the unit.
powerOnDelay	You can force the transceiver to inhibit the transmitted signal for this time after power on. Setting the value to zero disables this feature.
alarmThreshold	This is the threshold value for the TX level alarm.
suppressLevelAlarm	If set to ON, the transceiver does not report a TX level alarm.
txAutoOn	Defines the default TX ON/OFF state when the unit gets powered on.
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. The recommended line setting for the use with the satnms software is *9600,N,8,1* . See below how to change this setting at the transceiver.

Communication protocol / Interface type

This device driver requires the Terminal-CR-CRLF-E protocol. You can use either the transceiver's RS232 port or the RS485 interface, however, you must set the device address properly if you are using the latter.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerOnDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 600)
config.alarmThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 64)
config.suppressLevelAlarm	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.txAutoOn	CHOICE	SAVE SETUP	EnumRange (OFF ON)
info.revision	TEXT	R/O	StringRange R/O
info.rfband	TEXT	R/O	StringRange R/O
info.mode	TEXT	R/O	StringRange R/O
info.hpa	TEXT	R/O	StringRange R/O
info.synths	TEXT	R/O	StringRange R/O
info.maxpower	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.measuredLevel	INTEGER	R/O	IntegerRange R/O (0 .. 200)
rx.frequency	FLOAT		DoubleRange (0 .. 12750)
rx.gain	FLOAT		DoubleRange (-10 .. 5)
rx.on	CHOICE		EnumRange (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (-100.0 .. 100.0)
tx.frequency	FLOAT		DoubleRange (0 .. 14500)
tx.gain	FLOAT		DoubleRange (52 .. 62)

name	*type*	*flags*	*range*
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.state	HEX	R/O	HexRange R/O (0 .. 0)
internal.tx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.tx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmin	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
internal.rx.fmax	FLOAT	R/O	DoubleRange R/O (0 .. 40000)
address	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Supply +15V)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Supply +12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Supply +5V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Supply -12V)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Synth ID)
faults.06	ALARM	R/O	AlarmFlagRange R/O (A/D mux)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.09	ALARM	R/O	AlarmFlagRange R/O (RX level)
faults.10	ALARM	R/O	AlarmFlagRange R/O (RX synth lock)
faults.12	ALARM	R/O	AlarmFlagRange R/O (HPC level)
faults.13	ALARM	R/O	AlarmFlagRange R/O (UpConv level)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TX level)
faults.15	ALARM	R/O	AlarmFlagRange R/O (TX synth lock)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (TX power on delay)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TWT cycling)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TWT)

9.595 SVS-SWC100

This page describes the device driver and the device window for the SVS SWC 100 redundancy controller using SVS-Serial protocol.

It was tested with 1:1 Redundancy Controller

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The redundancy and switch parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal). The factory default is 1.
readBackDelay	Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Ethernet/Serial Interface

The Controller may be controlled through their Ethernet interface. The default port number for this is 4001. Assuming the controller is set to an IP address 10.0.0.1, you have to configure a communication interface "10.0.0.1:4001" int the sat-nms VLC setup in order to make the VLC talk to the controller. Beside this, the controller must be configured to the following serial settings at it's web-interface.

- Baudrate: 115200
- Parity: None
- Data Bits: 8
- Stop Bits: 1
- Flow Control: None
- FIFO: Enable
- Interface: RS485

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
position	CHOICE		EnumRange (A B)
protection	CHOICE		EnumRange (AUTO MANUAL)
selected	CHOICE	R/O	EnumRange R/O (A B)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch Fail A)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch Fail B)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Switch indication)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Communication Frame)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Communication Overrun)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Communication TX)

9.596 Sampler

The logical device Sampler monitors up to 32 parameters from other devices. Triggered by the user for one element (sample button) or for all elements together (sample all button) the device takes the current value of the monitored parameters.

This device is useful to get for example measurement value from a device to a given moment. After sampling a parameter you can use other devices to work with this information (e.g. File-Recorder).

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- List of sampled parameters
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance contains settings to configure this logical device.

parameter	***description***
input-xx.Id	Parameter IDs

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input-01.Id	TEXT	SAVE SETUP	StringRange
config.input-02.Id	TEXT	SAVE SETUP	StringRange
config.input-03.Id	TEXT	SAVE SETUP	StringRange
config.input-04.Id	TEXT	SAVE SETUP	StringRange
config.input-05.Id	TEXT	SAVE SETUP	StringRange
config.input-06.Id	TEXT	SAVE SETUP	StringRange
config.input-07.Id	TEXT	SAVE SETUP	StringRange
config.input-08.Id	TEXT	SAVE SETUP	StringRange
config.input-09.Id	TEXT	SAVE SETUP	StringRange
config.input-10.Id	TEXT	SAVE SETUP	StringRange
config.input-11.Id	TEXT	SAVE SETUP	StringRange
config.input-12.Id	TEXT	SAVE SETUP	StringRange
config.input-13.Id	TEXT	SAVE SETUP	StringRange
config.input-14.Id	TEXT	SAVE SETUP	StringRange
config.input-15.Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.input-16.Id	TEXT	SAVE SETUP	StringRange
config.input-17.Id	TEXT	SAVE SETUP	StringRange
config.input-18.Id	TEXT	SAVE SETUP	StringRange
config.input-19.Id	TEXT	SAVE SETUP	StringRange
config.input-20.Id	TEXT	SAVE SETUP	StringRange
config.input-21.Id	TEXT	SAVE SETUP	StringRange
config.input-22.Id	TEXT	SAVE SETUP	StringRange
config.input-23.Id	TEXT	SAVE SETUP	StringRange
config.input-24.Id	TEXT	SAVE SETUP	StringRange
config.input-25.Id	TEXT	SAVE SETUP	StringRange
config.input-26.Id	TEXT	SAVE SETUP	StringRange
config.input-27.Id	TEXT	SAVE SETUP	StringRange
config.input-28.Id	TEXT	SAVE SETUP	StringRange
config.input-29.Id	TEXT	SAVE SETUP	StringRange
config.input-30.Id	TEXT	SAVE SETUP	StringRange
config.input-31.Id	TEXT	SAVE SETUP	StringRange
config.input-32.Id	TEXT	SAVE SETUP	StringRange
sample.01	INTEGER		IntegerRange (1 .. 32)
sample.02	INTEGER		IntegerRange (1 .. 32)
sample.03	INTEGER		IntegerRange (1 .. 32)
sample.04	INTEGER		IntegerRange (1 .. 32)
sample.05	INTEGER		IntegerRange (1 .. 32)
sample.06	INTEGER		IntegerRange (1 .. 32)

name	*type*	*flags*	*range*
sample.07	INTEGER		IntegerRange (1 .. 32)
sample.08	INTEGER		IntegerRange (1 .. 32)
sample.09	INTEGER		IntegerRange (1 .. 32)
sample.10	INTEGER		IntegerRange (1 .. 32)
sample.11	INTEGER		IntegerRange (1 .. 32)
sample.12	INTEGER		IntegerRange (1 .. 32)
sample.13	INTEGER		IntegerRange (1 .. 32)
sample.14	INTEGER		IntegerRange (1 .. 32)
sample.15	INTEGER		IntegerRange (1 .. 32)
sample.16	INTEGER		IntegerRange (1 .. 32)
sample.17	INTEGER		IntegerRange (1 .. 32)
sample.18	INTEGER		IntegerRange (1 .. 32)
sample.19	INTEGER		IntegerRange (1 .. 32)
sample.20	INTEGER		IntegerRange (1 .. 32)
sample.21	INTEGER		IntegerRange (1 .. 32)
sample.22	INTEGER		IntegerRange (1 .. 32)

name	*type*	*flags*	*range*
sample.23	INTEGER		IntegerRange (1 .. 32)
sample.24	INTEGER		IntegerRange (1 .. 32)
sample.25	INTEGER		IntegerRange (1 .. 32)
sample.26	INTEGER		IntegerRange (1 .. 32)
sample.27	INTEGER		IntegerRange (1 .. 32)
sample.28	INTEGER		IntegerRange (1 .. 32)
sample.29	INTEGER		IntegerRange (1 .. 32)
sample.30	INTEGER		IntegerRange (1 .. 32)
sample.31	INTEGER		IntegerRange (1 .. 32)
sample.32	INTEGER		IntegerRange (1 .. 32)
sampleAll	TEXT		StringRange
input.01	TEXT		StringRange
input.02	TEXT		StringRange
input.03	TEXT		StringRange
input.04	TEXT		StringRange
input.05	TEXT		StringRange
input.06	TEXT		StringRange
input.07	TEXT		StringRange
input.08	TEXT		StringRange
input.09	TEXT		StringRange

name	*type*	*flags*	*range*
input.10	TEXT		StringRange
input.11	TEXT		StringRange
input.12	TEXT		StringRange
input.13	TEXT		StringRange
input.14	TEXT		StringRange
input.15	TEXT		StringRange
input.16	TEXT		StringRange
input.17	TEXT		StringRange
input.18	TEXT		StringRange
input.19	TEXT		StringRange
input.20	TEXT		StringRange
input.21	TEXT		StringRange
input.22	TEXT		StringRange
input.23	TEXT		StringRange
input.24	TEXT		StringRange
input.25	TEXT		StringRange
input.26	TEXT		StringRange
input.27	TEXT		StringRange
input.28	TEXT		StringRange
input.29	TEXT		StringRange
input.30	TEXT		StringRange
input.31	TEXT		StringRange
input.32	TEXT		StringRange
internal.01	TEXT		StringRange
internal.02	TEXT		StringRange
internal.03	TEXT		StringRange
internal.04	TEXT		StringRange

name	*type*	*flags*	*range*
internal.05	TEXT		StringRange
internal.06	TEXT		StringRange
internal.07	TEXT		StringRange
internal.08	TEXT		StringRange
internal.09	TEXT		StringRange
internal.10	TEXT		StringRange
internal.11	TEXT		StringRange
internal.12	TEXT		StringRange
internal.13	TEXT		StringRange
internal.14	TEXT		StringRange
internal.15	TEXT		StringRange
internal.16	TEXT		StringRange
internal.17	TEXT		StringRange
internal.18	TEXT		StringRange
internal.19	TEXT		StringRange
internal.20	TEXT		StringRange
internal.21	TEXT		StringRange
internal.22	TEXT		StringRange
internal.23	TEXT		StringRange
internal.24	TEXT		StringRange
internal.25	TEXT		StringRange
internal.26	TEXT		StringRange
internal.27	TEXT		StringRange
internal.28	TEXT		StringRange
internal.29	TEXT		StringRange
internal.30	TEXT		StringRange

name	*type*	*flags*	*range*
internal.31	TEXT		StringRange
internal.32	TEXT		StringRange

9.597 SatService-ACU-IDU

This page describes the device driver and the device window for the SatService ACU/IDU tracking controller (indoor unit). This driver allows the position and state of an sat-nms tracking unit to be monitored via RS232 from a M&C system or an VLC.

As the sat-nms ACU/IDU tracking controller provides it's own HTTP based user interface, it is recommended to install a link to the ACU user interface at the main task oriented screen of the application. The Frame Button may be used to provide this function.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing. No changes can be made with this version of the device driver.
- --- The second parameter page displays the tracking mode / state. Refer to the Antenna-Tracking manual page for an explanation of the tracking modes in particular.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

There are no special configuration settings necessary to communicate to the***satnms*** ACU indoor Unit. However, remember to set the line parameters for the serial interface to the correct values: *19200,N,8,1* .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
azimuth	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
elevation	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
polarization	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Out door unit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tracking)

9.598 SatService-ACU-ODM-Serial

This page describes the device driver and the device window for the sat-nms ACU outdoor module. The driver treats the ACU ODM as a plain antenna positioner, communicating via Serial Interface to it. The configuration parameters of the ACU ODM are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the M&C device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	n/a
timeSync	Enables or disables the date/time synchronization of the ACU-ODM with the MNC/ACU-IDU system time.
Serial	Remember to set the line parameters for the serial interface connected to the ACU-ODM. The RS232 interface of sat-nms ACU-ODM always operates at 9600 baud, no parity, 8 data bits, one stop bit. *9600,8,N,1* .

Serial protocol selection Depending on the device address set, the sat-nms ACU-ODM either runs framed protocol with start/stop characters and checksum or it provides a dumb terminal interface. If the address is set in the ACU configuration (web page) to 'A' .. 'G' use Miteq-MOD95 protocol, if the address is 'NONE' select Terminal-CR-CRLF protocol.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU Outdoor Module. The ACU-ODM will update its internal clock.

The ACU-ODM does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU-ODM clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU-ODM updates its internal clock. Use the web browser interface to check the current date and time of the ACU-ODM.

This feature syncs the date and time between ACU-ODM and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)

name	*type*	*flags*	*range*
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.loadList	TEXT	NOPRESET	StringRange
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)

9.599 SatService-ACU-ODM

This page describes the device driver and the device window for the sat-nms ACU outdoor module. The driver treats the ACU ODM as a plain antenna positioner, communicating via Ethernet/HTTP to it. The configuration parameters of the ACU ODM are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the M&C device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*

parameter	*description*
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".
timeSync	Enables or disables the date/time synchronization of the ACU-ODM with the MNC/ACU-IDU system time.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU Outdoor Module. The ACU-ODM will update its internal clock.

The ACU-ODM does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU-ODM clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU-ODM updates its internal clock. Use the web browser interface to check the current date and time of the ACU-ODM.

This feature syncs the date and time between ACU-ODM and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.mode1	TEXT	R/O	StringRange R/O
state.mode4	GENERIC	R/O	ObjectRange R/O generic object
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS

name	*type*	*flags*	*range*
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)

name	*type*	*flags*	*range*
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)

9.600 SatService-ACU-Prediction

This page describes the *SatService-ACU-Prediction* device. This device is a supplement to the SatService-ACU-ODM device driver for the sat-nms ACU outdoor module.

The *SatService-ACU-Prediction* device adds access to the ACU's feature to calculate a 100 datapoint table from a given I11 or TLE dataset stored in the ACU's memory. This requires an ACU2 outdoor module (software version 3.0 or higher). The *SatService-ACU-Prediction* device works like a separate device driver which talks to the ACU-ODM in parallel to the basic SatService-ACU-ODM driver. The *SatService-ACU-Prediction* device may be configured in the same or in a different device thread (aka. INTERFACE), the HTTP protocol used does not

demand any restriction on this.

To use the *SatService-ACU-Prediction* device, select the Type of ephemerides to use, the memory number, start time and time increment. Finally click "SUBMIT" and the result will be shown.

Using the device in an automated or remote controlled environment works exactly in the same way: Set the variables 'ephemerisMode', 'memoryNumber', 'startTime' and 'timeIncrement' to specify the parameters of the request. Finally set the 'predict' variable to an arbitrary value to send the request to the ACU-ODM

The 100 data points then can be read from the 'result.nn' variables, where 'nn' is in the range of '00' to '99'. The variable 'state' holds information about success or failure of the operation. If the request failed, all 'result.nn' variables contain empty strings and the 'state' variable tells about the failure's reason. Please give the device about one second time to answer after sending the request and before you read the result strings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls request parameters and the result.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there is one configuration parameter which must be set to make the software talk to the antenna controller.

parameter	***description***
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
ephemerisType	CHOICE		EnumRange (TLE I11)
memoryNumber	INTEGER		IntegerRange (1 .. 99)
selectedTLE	CHOICE		EnumRange (00 - NONE)
selectedI11	CHOICE		EnumRange DIS (00 - NONE)
startTime	TEXT		StringRange
timeIncrement	FLOAT		DoubleRange (0.01 .. 24.00)
predict	TEXT	NOPRESET	StringRange
result	TEXT	R/O	StringRange R/O
state	TEXT	R/O	StringRange R/O
result.00	TEXT	R/O	StringRange R/O
result.01	TEXT	R/O	StringRange R/O
result.02	TEXT	R/O	StringRange R/O
result.03	TEXT	R/O	StringRange R/O
result.04	TEXT	R/O	StringRange R/O
result.05	TEXT	R/O	StringRange R/O
result.06	TEXT	R/O	StringRange R/O
result.07	TEXT	R/O	StringRange R/O
result.08	TEXT	R/O	StringRange R/O
result.09	TEXT	R/O	StringRange R/O
result.10	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.11	TEXT	R/O	StringRange R/O
result.12	TEXT	R/O	StringRange R/O
result.13	TEXT	R/O	StringRange R/O
result.14	TEXT	R/O	StringRange R/O
result.15	TEXT	R/O	StringRange R/O
result.16	TEXT	R/O	StringRange R/O
result.17	TEXT	R/O	StringRange R/O
result.18	TEXT	R/O	StringRange R/O
result.19	TEXT	R/O	StringRange R/O
result.20	TEXT	R/O	StringRange R/O
result.21	TEXT	R/O	StringRange R/O
result.22	TEXT	R/O	StringRange R/O
result.23	TEXT	R/O	StringRange R/O
result.24	TEXT	R/O	StringRange R/O
result.25	TEXT	R/O	StringRange R/O
result.26	TEXT	R/O	StringRange R/O
result.27	TEXT	R/O	StringRange R/O
result.28	TEXT	R/O	StringRange R/O
result.29	TEXT	R/O	StringRange R/O
result.30	TEXT	R/O	StringRange R/O
result.31	TEXT	R/O	StringRange R/O
result.32	TEXT	R/O	StringRange R/O
result.33	TEXT	R/O	StringRange R/O
result.34	TEXT	R/O	StringRange R/O
result.35	TEXT	R/O	StringRange R/O
result.36	TEXT	R/O	StringRange R/O
result.37	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.38	TEXT	R/O	StringRange R/O
result.39	TEXT	R/O	StringRange R/O
result.40	TEXT	R/O	StringRange R/O
result.41	TEXT	R/O	StringRange R/O
result.42	TEXT	R/O	StringRange R/O
result.43	TEXT	R/O	StringRange R/O
result.44	TEXT	R/O	StringRange R/O
result.45	TEXT	R/O	StringRange R/O
result.46	TEXT	R/O	StringRange R/O
result.47	TEXT	R/O	StringRange R/O
result.48	TEXT	R/O	StringRange R/O
result.49	TEXT	R/O	StringRange R/O
result.50	TEXT	R/O	StringRange R/O
result.51	TEXT	R/O	StringRange R/O
result.52	TEXT	R/O	StringRange R/O
result.53	TEXT	R/O	StringRange R/O
result.54	TEXT	R/O	StringRange R/O
result.55	TEXT	R/O	StringRange R/O
result.56	TEXT	R/O	StringRange R/O
result.57	TEXT	R/O	StringRange R/O
result.58	TEXT	R/O	StringRange R/O
result.59	TEXT	R/O	StringRange R/O
result.60	TEXT	R/O	StringRange R/O
result.61	TEXT	R/O	StringRange R/O
result.62	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.63	TEXT	R/O	StringRange R/O
result.64	TEXT	R/O	StringRange R/O
result.65	TEXT	R/O	StringRange R/O
result.66	TEXT	R/O	StringRange R/O
result.67	TEXT	R/O	StringRange R/O
result.68	TEXT	R/O	StringRange R/O
result.69	TEXT	R/O	StringRange R/O
result.70	TEXT	R/O	StringRange R/O
result.71	TEXT	R/O	StringRange R/O
result.72	TEXT	R/O	StringRange R/O
result.73	TEXT	R/O	StringRange R/O
result.74	TEXT	R/O	StringRange R/O
result.75	TEXT	R/O	StringRange R/O
result.76	TEXT	R/O	StringRange R/O
result.77	TEXT	R/O	StringRange R/O
result.78	TEXT	R/O	StringRange R/O
result.79	TEXT	R/O	StringRange R/O
result.80	TEXT	R/O	StringRange R/O
result.81	TEXT	R/O	StringRange R/O
result.82	TEXT	R/O	StringRange R/O
result.83	TEXT	R/O	StringRange R/O
result.84	TEXT	R/O	StringRange R/O
result.85	TEXT	R/O	StringRange R/O
result.86	TEXT	R/O	StringRange R/O
result.87	TEXT	R/O	StringRange R/O
result.88	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
result.89	TEXT	R/O	StringRange R/O
result.90	TEXT	R/O	StringRange R/O
result.91	TEXT	R/O	StringRange R/O
result.92	TEXT	R/O	StringRange R/O
result.93	TEXT	R/O	StringRange R/O
result.94	TEXT	R/O	StringRange R/O
result.95	TEXT	R/O	StringRange R/O
result.96	TEXT	R/O	StringRange R/O
result.97	TEXT	R/O	StringRange R/O
result.98	TEXT	R/O	StringRange R/O
result.99	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange

9.601 SatService-ACU2-Serial

This page describes the device driver and the device window for the sat-nms ACU2 antenna controller. This driver uses the Serial Interface for communication with it. The configuration parameters of the ACU2 are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the MNC device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Show the tracking state.
- --- Target list to load, save and delete targets on the ACU
- --- Set the tracking parameters
- --- Editor for TLE's parameters
- --- Editor for I11 (Intelsat) Parameters.
- --- Target Editor
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	n/a
timeSync	Enables or disables the date/time synchronization of the ACU with the MNC/ACU-IDU system time.
Serial	Remember to set the line parameters for the serial interface connected to the ACU. The RS232 interface of sat-nms ACU always operates at 9600 baud, no parity, 8 data bits, one stop bit. Defaults is *9600,8,N,1* .

Serial protocol selection Depending on the device address set, the sat-nms ACU either runs framed protocol with start/stop characters and checksum or it provides a dumb terminal interface. If the address is set in the ACU configuration (web page) to 'A' .. 'G' use Miteq-MOD95 protocol, if the address is 'NONE' select Terminal-CR-CRLF protocol.

Please remember to set the baudrate on the ACU.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU . The ACU will update its internal clock.

The ACU does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU updates its internal clock. Use the web browser interface to check the current date and time of the ACU.

This feature syncs the date and time between ACU and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.i11tle	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.4axes	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux2	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.obi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.target	TEXT	R/O	StringRange R/O
internal.targetNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
pl2StepEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.bcrx.polAlias	TEXT	SETUP	StringRange
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.pl2LoLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.pl2HiLimit	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.targetTainted	TEXT	R/O	StringRange R/O
state.taintedText	TEXT	R/O	StringRange R/O
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (0 .. 0)
trackMode	CHOICE		EnumRange (UNKNOWN OFF STEP ADAPTIVE)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)

name	*type*	*flags*	*range*
levelZero	TEXT	NOPRESET	StringRange DIS
spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5)
spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)
azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
polPredict	CHOICE		EnumRange DIS (OFF ON)
inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
clearTainted	TEXT	NOPRESET	StringRange
bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)
bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)
bcrx.polarization	CHOICE		EnumRange DIS (H V)
bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)

name	*type*	*flags*	*range*
bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)
saveMode	CHOICE		EnumRange (ALL NO-POINTING)
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.loadList	TEXT	NOPRESET	StringRange
target.needUpdate	BOOLEAN	NOPRESET	BooleanRange (true false)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.5	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.6	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.7	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.8	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
aux.out.9	INTEGER		IntegerRange DIS (0 .. 1)
aux.out.10	INTEGER		IntegerRange DIS (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TLE data age)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TLE data parse)
faults.27	ALARM	R/O	AlarmFlagRange R/O (I11 data age)
faults.28	ALARM	R/O	AlarmFlagRange R/O (I11 data parse)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)
faults.30	ALARM	R/O	AlarmFlagRange R/O (PL2 high limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PL2 low limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (PL2 motor fault)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PL2 motor timeout)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PL2 motor stopped)

9.602 SatService-ACU2

This page describes the device driver and the device window for the sat-nms ACU2 antenna controller. This driver uses Ethernet/HTTP for communication with it. The configuration parameters of the ACU2 are accessible through the ACU web based user interface, they are not replicated as configuration parameters in the MNC device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main parameter page displays/controls the antenna pointing. Actual readings and set points are shown in separate fields, so you can see where the antenna is going to while it is moving.
- --- Show the tracking state.
- --- Target list to load, save and delete targets on the ACU
- --- Set the tracking parameters
- --- Editor for TLE's parameters
- --- Editor for I11 (Intelsat) Parameters.
- --- Target Editor
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".
timeSync	Enables or disables the date/time synchronization of the ACU with the MNC/ACU-IDU system time.

Time Synchronization

If enabled via the setup option described above, the M&C System or ACU Indoor Unit (running this drivers) will send date and time in regular intervals to the ACU . The ACU will update its internal clock.

The ACU does not have a time zone configuration, so it simply uses the time send by the MNC/ACU-IDU. For example: If the system clock of the MNC/ACU-IDU runs on UTC, the ACU clock is also set to UTC

Do not enable time synchronization for the first time if you just doing step or adaptive tracking, because a large leap in time will produce a wrong tracking model. After first synchronization you can re-enable step or adaptive tracking again. It need approx. 40 seconds after enabling until the ACU updates its internal clock. Use the web browser interface to check the current date and time of the ACU.

This feature syncs the date and time between ACU and MNC/ACU-IDU. If you want to synchronize the date and time of the MNC/ACU-IDU server you can use an external NTP server. Please refer to the sat-nms M&C Installation Manual.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.caps.tracking	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.polar	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.memtrack	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.clswitch	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.rellevel	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.200targets	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.i11tle	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.4axes	BOOLEAN	R/O	BooleanRange R/O (true false)
info.caps.mhyst	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.poslog	TEXT	R/O	StringRange R/O
info.beaconType	CHOICE		EnumRange (SATNMS VOLTAGE)
internal.azLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.azHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.elHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plLoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.plHiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl2LoLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl2HiLimit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ibit	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.ibitaux	HEX	R/O	HexRange R/O (0 .. 0)
internal.obit	HEX	R/O	HexRange R/O (0 .. 0)
internal.obi2	HEX	R/O	HexRange R/O (0 .. 0)
internal.tflt	HEX	R/O	HexRange R/O (0 .. 0)
internal.pcnt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pcnt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.time	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.yy	TEXT	R/O	StringRange R/O
internal.mo	TEXT	R/O	StringRange R/O
internal.dd	TEXT	R/O	StringRange R/O
internal.hh	TEXT	R/O	StringRange R/O
internal.mi	TEXT	R/O	StringRange R/O
internal.ss	TEXT	R/O	StringRange R/O
internal.state	TEXT	R/O	StringRange R/O
internal.sleep	TEXT	R/O	StringRange R/O
internal.cmod	TEXT	R/O	StringRange R/O DIS
initialized	TEXT	R/O	StringRange R/O
pl2StepEnable	BOOLEAN	R/O	BooleanRange R/O (true false)
config.timeSync	CHOICE	SAVE SETUP	EnumRange (ENABLED DISABLED)
config.bcrx.polAlias	TEXT	SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
clswitch	CHOICE		EnumRange DIS (LINEAR CIRCULAR UNKNOWN)
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange DIS
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.cl	TEXT	R/O	StringRange R/O DIS
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
state.stopped	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.absLevel	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode4	GENERIC	R/O	ObjectRange R/O generic object
state.azJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elJitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.azAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.elAmplitude	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.trackHours	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.azModel	TEXT	R/O	StringRange R/O
state.elModel	TEXT	R/O	StringRange R/O
state.modelAge	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.targetTainted	TEXT	R/O	StringRange R/O
state.taintedText	TEXT	R/O	StringRange R/O
state.selectedTarget	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
trackMode	CHOICE		EnumRange (OFF STEP ADAPTIVE)
initMode	CHOICE		EnumRange DIS (STORED-POSITION ORBIT I11 TLE MODEL)
interval	INTEGER		IntegerRange DIS (1 .. 1638)
stepSize	INTEGER		IntegerRange DIS (1 .. 100)
measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
averaging	INTEGER		IntegerRange DIS (1 .. 25)
frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
smoothing	INTEGER		IntegerRange DIS (0 .. 6)
azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
levelZero	TEXT	NOPRESET	StringRange DIS
spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5)
spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)
modelHysteresis	INTEGER		IntegerRange DIS (0 .. 100)
modelBeforeTrack	CHOICE		EnumRange DIS (OFF ON)
azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)

name	*type*	*flags*	*range*
elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
selectedTLE	CHOICE		EnumRange DIS (00 - NONE)
selectedI11	CHOICE		EnumRange DIS (00 - NONE)
tleNo	INTEGER		IntegerRange DIS (0 .. 99)
i11No	INTEGER		IntegerRange DIS (0 .. 99)
maxAge	FLOAT		DoubleRange DIS (0.0 .. 999.0)
retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
polPredict	CHOICE		EnumRange DIS (OFF ON)
inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
reset	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
motorReset	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
logPosition	TEXT	NOPRESET	StringRange
clearTainted	TEXT	NOPRESET	StringRange
bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)
bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)

name	*type*	*flags*	*range*
bcrx.polarization	CHOICE		EnumRange DIS (H V)
bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)
bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)
edit.tle.load	CHOICE		EnumRange DIS (00 - NONE)
edit.tle.no	INTEGER		IntegerRange DIS (1 .. 99)
edit.tle.save	TEXT	NOPRESET	StringRange DIS
edit.tle.destination	INTEGER		IntegerRange DIS (1 .. 99)
edit.tle.name	TEXT		StringRange DIS
edit.tle.line1	TEXT		StringRange DIS
edit.tle.line2	TEXT		StringRange DIS
edit.i11.load	CHOICE		EnumRange DIS (00 - NONE)
edit.i11.no	INTEGER		IntegerRange DIS (1 .. 99)
edit.i11.save	TEXT	NOPRESET	StringRange DIS

name	*type*	*flags*	*range*
edit.i11.destination	INTEGER		IntegerRange DIS (1 .. 99)
edit.i11.name	TEXT		StringRange DIS
edit.i11.epoch	TEXT		StringRange DIS
edit.i11.lm0	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lm1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lm2	FLOAT		DoubleRange DIS (0.000000 .. 0.000000)
edit.i11.lonc	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lonc1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lons	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lons1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.latc	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.latc1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lats	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.lats1	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.predlon	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.i11.predlat	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
edit.target.load	CHOICE		EnumRange DIS (000 - ADAPTIVE TRACKING MEMORY)
edit.target.no	INTEGER		IntegerRange DIS (1 .. 199)

name	*type*	*flags*	*range*
edit.target.save	TEXT	NOPRESET	StringRange DIS
edit.target.destination	INTEGER		IntegerRange DIS (1 .. 199)
edit.target.selectedTLE	CHOICE		EnumRange DIS (00 - NONE)
edit.target.selectedI11	CHOICE		EnumRange DIS (00 - NONE)
edit.target.tleNo	INTEGER		IntegerRange DIS (0 .. 99)
edit.target.i11No	INTEGER		IntegerRange DIS (0 .. 99)
edit.target.trackMode	CHOICE		EnumRange DIS (OFF STEP ADAPTIVE MEMORY STEP-TLE STEP-I11 TLE I11)
edit.target.initMode	CHOICE		EnumRange DIS (STORED-POSITION ORBIT I11 TLE)
edit.target.interval	INTEGER		IntegerRange DIS (60 .. 1638)
edit.target.stepSize	INTEGER		IntegerRange DIS (1 .. 100)
edit.target.measDelay	INTEGER		IntegerRange DIS (1 .. 9999)
edit.target.recoveryDelay	INTEGER		IntegerRange DIS (100 .. 30000)
edit.target.averaging	INTEGER		IntegerRange DIS (1 .. 25)
edit.target.levelThreshold	FLOAT		DoubleRange DIS (-999.00 .. 100.00)
edit.target.jitterThreshold	INTEGER		IntegerRange DIS (0 .. 100)
edit.target.smoothing	INTEGER		IntegerRange DIS (0 .. 6)
edit.target.azMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
edit.target.elMaxModel	CHOICE		EnumRange DIS (SMALL MEDIUM LARGE)
edit.target.levelOffset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
edit.target.spindleSaveMode	CHOICE		EnumRange DIS (OFF 1 2 3 4 5 6 7 8 9 10 11 12)
edit.target.spindleSaveThresh	INTEGER		IntegerRange DIS (0 .. 100)

name	*type*	*flags*	*range*
edit.target.modelHysteresis	INTEGER		IntegerRange DIS (0 .. 100)
edit.target.modelBeforeTrack	CHOICE		EnumRange DIS (OFF ON)
edit.target.azOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
edit.target.elOffset	FLOAT		DoubleRange DIS (-180.000 .. 180.000)
edit.target.orbit	FLOAT		DoubleRange DIS (0.000 .. 0.000)
edit.target.maxAge	FLOAT		DoubleRange DIS (0.0 .. 999.0)
edit.target.polPredict	CHOICE		EnumRange DIS (OFF ON)
edit.target.inclination	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
edit.target.satPolOffset	FLOAT		DoubleRange DIS (-45.000 .. 45.000)
edit.target.retryOnFault	CHOICE		EnumRange DIS (NEVER ONCE FOREVER)
edit.target.azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
edit.target.elevation	FLOAT		DoubleRange (0.000 .. 0.000)
edit.target.polarization	FLOAT		DoubleRange (0.00 .. 0.00)
edit.target.polarization2	FLOAT		DoubleRange DIS (0.00 .. 0.00)
edit.target.name	TEXT		StringRange
edit.target.frequency	FLOAT		DoubleRange (1000.000 .. 40000.000)
edit.target.bcrx.attenuation	CHOICE		EnumRange DIS (30 dB 20 dB 10 dB 0 dB)
edit.target.bcrx.averaging	CHOICE		EnumRange DIS (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
edit.target.bcrx.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)

name	*type*	*flags*	*range*
edit.target.bcrx.bandwidth	CHOICE		EnumRange DIS (6 kHz 12 kHz 30 kHz 100 kHz)
edit.target.bcrx.polarization	CHOICE		EnumRange DIS (H V)
edit.target.bcrx.threshold	FLOAT		DoubleRange DIS (-999.99 .. 0.00)
edit.target.bcrx.speccomp	CHOICE		EnumRange DIS (OFF)
edit.target.bcrx.ftrack.on	CHOICE		EnumRange DIS (OFF ON)
edit.target.bcrx.ftrack.width	INTEGER		IntegerRange DIS (10 .. 1000)
edit.target.bcrx.ftrack.interval	INTEGER		IntegerRange DIS (1 .. 21600)
edit.target.bcrx.ssrch.delay	INTEGER		IntegerRange DIS (0 .. 600)
edit.target.bcrx.ssrch.enable	CHOICE		EnumRange DIS (OFF ON)
edit.target.bcrx.cn.on	CHOICE		EnumRange DIS (OFF C/N C/NO)
edit.target.bcrx.cn.frequency	FLOAT		DoubleRange DIS (950.000 .. 40000.000)
edit.target.bcrx.cn.interval	INTEGER		IntegerRange DIS (0 .. 0)
edit.target.bcrx.offset	FLOAT		DoubleRange DIS (0.00 .. 0.00)
edit.target.bcrx.output	FLOAT		DoubleRange DIS (0.000 .. 0.000)
saveMode	CHOICE		EnumRange (ALL NO-POINTING)
initialPointingOverride	CHOICE		EnumRange (STORED-POSITION ORBIT TLE I11 MODEL FROM-TARGET)
target.name	TEXT		StringRange
target.0	TEXT	R/O	StringRange R/O
target.1	TEXT	R/O	StringRange R/O
target.2	TEXT	R/O	StringRange R/O
target.3	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.4	TEXT	R/O	StringRange R/O
target.5	TEXT	R/O	StringRange R/O
target.6	TEXT	R/O	StringRange R/O
target.7	TEXT	R/O	StringRange R/O
target.8	TEXT	R/O	StringRange R/O
target.9	TEXT	R/O	StringRange R/O
target.10	TEXT	R/O	StringRange R/O
target.11	TEXT	R/O	StringRange R/O
target.12	TEXT	R/O	StringRange R/O
target.13	TEXT	R/O	StringRange R/O
target.14	TEXT	R/O	StringRange R/O
target.15	TEXT	R/O	StringRange R/O
target.16	TEXT	R/O	StringRange R/O
target.17	TEXT	R/O	StringRange R/O
target.18	TEXT	R/O	StringRange R/O
target.19	TEXT	R/O	StringRange R/O
target.20	TEXT	R/O	StringRange R/O
target.21	TEXT	R/O	StringRange R/O
target.22	TEXT	R/O	StringRange R/O
target.23	TEXT	R/O	StringRange R/O
target.24	TEXT	R/O	StringRange R/O
target.25	TEXT	R/O	StringRange R/O
target.26	TEXT	R/O	StringRange R/O
target.27	TEXT	R/O	StringRange R/O
target.28	TEXT	R/O	StringRange R/O
target.29	TEXT	R/O	StringRange R/O
target.30	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.31	TEXT	R/O	StringRange R/O
target.32	TEXT	R/O	StringRange R/O
target.33	TEXT	R/O	StringRange R/O
target.34	TEXT	R/O	StringRange R/O
target.35	TEXT	R/O	StringRange R/O
target.36	TEXT	R/O	StringRange R/O
target.37	TEXT	R/O	StringRange R/O
target.38	TEXT	R/O	StringRange R/O
target.39	TEXT	R/O	StringRange R/O
target.40	TEXT	R/O	StringRange R/O
target.41	TEXT	R/O	StringRange R/O
target.42	TEXT	R/O	StringRange R/O
target.43	TEXT	R/O	StringRange R/O
target.44	TEXT	R/O	StringRange R/O
target.45	TEXT	R/O	StringRange R/O
target.46	TEXT	R/O	StringRange R/O
target.47	TEXT	R/O	StringRange R/O
target.48	TEXT	R/O	StringRange R/O
target.49	TEXT	R/O	StringRange R/O
target.50	TEXT	R/O	StringRange R/O
target.51	TEXT	R/O	StringRange R/O
target.52	TEXT	R/O	StringRange R/O
target.53	TEXT	R/O	StringRange R/O
target.54	TEXT	R/O	StringRange R/O
target.55	TEXT	R/O	StringRange R/O
target.56	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.57	TEXT	R/O	StringRange R/O
target.58	TEXT	R/O	StringRange R/O
target.59	TEXT	R/O	StringRange R/O
target.60	TEXT	R/O	StringRange R/O
target.61	TEXT	R/O	StringRange R/O
target.62	TEXT	R/O	StringRange R/O
target.63	TEXT	R/O	StringRange R/O
target.64	TEXT	R/O	StringRange R/O
target.65	TEXT	R/O	StringRange R/O
target.66	TEXT	R/O	StringRange R/O
target.67	TEXT	R/O	StringRange R/O
target.68	TEXT	R/O	StringRange R/O
target.69	TEXT	R/O	StringRange R/O
target.70	TEXT	R/O	StringRange R/O
target.71	TEXT	R/O	StringRange R/O
target.72	TEXT	R/O	StringRange R/O
target.73	TEXT	R/O	StringRange R/O
target.74	TEXT	R/O	StringRange R/O
target.75	TEXT	R/O	StringRange R/O
target.76	TEXT	R/O	StringRange R/O
target.77	TEXT	R/O	StringRange R/O
target.78	TEXT	R/O	StringRange R/O
target.79	TEXT	R/O	StringRange R/O
target.80	TEXT	R/O	StringRange R/O
target.81	TEXT	R/O	StringRange R/O
target.82	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.83	TEXT	R/O	StringRange R/O
target.84	TEXT	R/O	StringRange R/O
target.85	TEXT	R/O	StringRange R/O
target.86	TEXT	R/O	StringRange R/O
target.87	TEXT	R/O	StringRange R/O
target.88	TEXT	R/O	StringRange R/O
target.89	TEXT	R/O	StringRange R/O
target.90	TEXT	R/O	StringRange R/O
target.91	TEXT	R/O	StringRange R/O
target.92	TEXT	R/O	StringRange R/O
target.93	TEXT	R/O	StringRange R/O
target.94	TEXT	R/O	StringRange R/O
target.95	TEXT	R/O	StringRange R/O
target.96	TEXT	R/O	StringRange R/O
target.97	TEXT	R/O	StringRange R/O
target.98	TEXT	R/O	StringRange R/O
target.99	TEXT	R/O	StringRange R/O
target.100	TEXT	R/O	StringRange R/O
target.101	TEXT	R/O	StringRange R/O
target.102	TEXT	R/O	StringRange R/O
target.103	TEXT	R/O	StringRange R/O
target.104	TEXT	R/O	StringRange R/O
target.105	TEXT	R/O	StringRange R/O
target.106	TEXT	R/O	StringRange R/O
target.107	TEXT	R/O	StringRange R/O
target.108	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.109	TEXT	R/O	StringRange R/O
target.110	TEXT	R/O	StringRange R/O
target.111	TEXT	R/O	StringRange R/O
target.112	TEXT	R/O	StringRange R/O
target.113	TEXT	R/O	StringRange R/O
target.114	TEXT	R/O	StringRange R/O
target.115	TEXT	R/O	StringRange R/O
target.116	TEXT	R/O	StringRange R/O
target.117	TEXT	R/O	StringRange R/O
target.118	TEXT	R/O	StringRange R/O
target.119	TEXT	R/O	StringRange R/O
target.120	TEXT	R/O	StringRange R/O
target.121	TEXT	R/O	StringRange R/O
target.122	TEXT	R/O	StringRange R/O
target.123	TEXT	R/O	StringRange R/O
target.124	TEXT	R/O	StringRange R/O
target.125	TEXT	R/O	StringRange R/O
target.126	TEXT	R/O	StringRange R/O
target.127	TEXT	R/O	StringRange R/O
target.128	TEXT	R/O	StringRange R/O
target.129	TEXT	R/O	StringRange R/O
target.130	TEXT	R/O	StringRange R/O
target.131	TEXT	R/O	StringRange R/O
target.132	TEXT	R/O	StringRange R/O
target.133	TEXT	R/O	StringRange R/O
target.134	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.135	TEXT	R/O	StringRange R/O
target.136	TEXT	R/O	StringRange R/O
target.137	TEXT	R/O	StringRange R/O
target.138	TEXT	R/O	StringRange R/O
target.139	TEXT	R/O	StringRange R/O
target.140	TEXT	R/O	StringRange R/O
target.141	TEXT	R/O	StringRange R/O
target.142	TEXT	R/O	StringRange R/O
target.143	TEXT	R/O	StringRange R/O
target.144	TEXT	R/O	StringRange R/O
target.145	TEXT	R/O	StringRange R/O
target.146	TEXT	R/O	StringRange R/O
target.147	TEXT	R/O	StringRange R/O
target.148	TEXT	R/O	StringRange R/O
target.149	TEXT	R/O	StringRange R/O
target.150	TEXT	R/O	StringRange R/O
target.151	TEXT	R/O	StringRange R/O
target.152	TEXT	R/O	StringRange R/O
target.153	TEXT	R/O	StringRange R/O
target.154	TEXT	R/O	StringRange R/O
target.155	TEXT	R/O	StringRange R/O
target.156	TEXT	R/O	StringRange R/O
target.157	TEXT	R/O	StringRange R/O
target.158	TEXT	R/O	StringRange R/O
target.159	TEXT	R/O	StringRange R/O
target.160	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.161	TEXT	R/O	StringRange R/O
target.162	TEXT	R/O	StringRange R/O
target.163	TEXT	R/O	StringRange R/O
target.164	TEXT	R/O	StringRange R/O
target.165	TEXT	R/O	StringRange R/O
target.166	TEXT	R/O	StringRange R/O
target.167	TEXT	R/O	StringRange R/O
target.168	TEXT	R/O	StringRange R/O
target.169	TEXT	R/O	StringRange R/O
target.170	TEXT	R/O	StringRange R/O
target.171	TEXT	R/O	StringRange R/O
target.172	TEXT	R/O	StringRange R/O
target.173	TEXT	R/O	StringRange R/O
target.174	TEXT	R/O	StringRange R/O
target.175	TEXT	R/O	StringRange R/O
target.176	TEXT	R/O	StringRange R/O
target.177	TEXT	R/O	StringRange R/O
target.178	TEXT	R/O	StringRange R/O
target.179	TEXT	R/O	StringRange R/O
target.180	TEXT	R/O	StringRange R/O
target.181	TEXT	R/O	StringRange R/O
target.182	TEXT	R/O	StringRange R/O
target.183	TEXT	R/O	StringRange R/O
target.184	TEXT	R/O	StringRange R/O
target.185	TEXT	R/O	StringRange R/O
target.186	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
target.187	TEXT	R/O	StringRange R/O
target.188	TEXT	R/O	StringRange R/O
target.189	TEXT	R/O	StringRange R/O
target.190	TEXT	R/O	StringRange R/O
target.191	TEXT	R/O	StringRange R/O
target.192	TEXT	R/O	StringRange R/O
target.193	TEXT	R/O	StringRange R/O
target.194	TEXT	R/O	StringRange R/O
target.195	TEXT	R/O	StringRange R/O
target.196	TEXT	R/O	StringRange R/O
target.197	TEXT	R/O	StringRange R/O
target.198	TEXT	R/O	StringRange R/O
target.199	TEXT	R/O	StringRange R/O
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.save	INTEGER	NOPRESET	IntegerRange (0 .. 99)
target.delete	INTEGER	NOPRESET	IntegerRange (0 .. 99)
goto	INTEGER	NOPRESET	IntegerRange (0 .. 99)
grmc	TEXT	R/O	StringRange R/O
aux.in.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
aux.in.5	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.6	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
aux.in.7	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.in.8	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
aux.out.1	INTEGER		IntegerRange (0 .. 1)
aux.out.2	INTEGER		IntegerRange (0 .. 1)
aux.out.3	INTEGER		IntegerRange (0 .. 1)
aux.out.4	INTEGER		IntegerRange (0 .. 1)
aux.out.5	INTEGER		IntegerRange (0 .. 1)
aux.out.6	INTEGER		IntegerRange (0 .. 1)
aux.out.7	INTEGER		IntegerRange (0 .. 1)
aux.out.8	INTEGER		IntegerRange (0 .. 1)
aux.out.9	INTEGER		IntegerRange DIS (0 .. 1)
aux.out.10	INTEGER		IntegerRange DIS (0 .. 1)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ high limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AZ low limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EL high limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EL low limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PL high limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PL low limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ motor fault)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ motor timeout)
faults.09	ALARM	R/O	AlarmFlagRange R/O (EL motor fault)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (EL motor timeout)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PL motor fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PL motor timeout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (AZ motor stopped)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL motor stopped)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PL motor stopped)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Beacon level)
faults.17	ALARM	R/O	AlarmFlagRange R/O (AZ peaking)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL peaking)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Model fit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Model reset)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Hub/Limit fault)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Cabinet open/Handheld)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TLE data age)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (TLE data parse)
faults.27	ALARM	R/O	AlarmFlagRange R/O (I11 data age)
faults.28	ALARM	R/O	AlarmFlagRange R/O (I11 data parse)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Soft Limits)
faults.30	ALARM	R/O	AlarmFlagRange R/O (PL2 high limit)
faults.31	ALARM	R/O	AlarmFlagRange R/O (PL2 low limit)
faults.32	ALARM	R/O	AlarmFlagRange R/O (PL2 motor fault)
faults.33	ALARM	R/O	AlarmFlagRange R/O (PL2 motor timeout)
faults.34	ALARM	R/O	AlarmFlagRange R/O (PL2 motor stopped)

9.603 SatService-Beacon-Receiver

This page describes the device driver and the device window for the sat-nms LBRX beacon receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the receive level and lets you set the basic receive parameters like frequency, bandwidth and input attenuation.
- --- This page lets you configure the C/N measurement mode of the receiver and displays the signal C/N if this mode is active.
- --- This page contains the parameters to configure the frequency tracking facility of the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

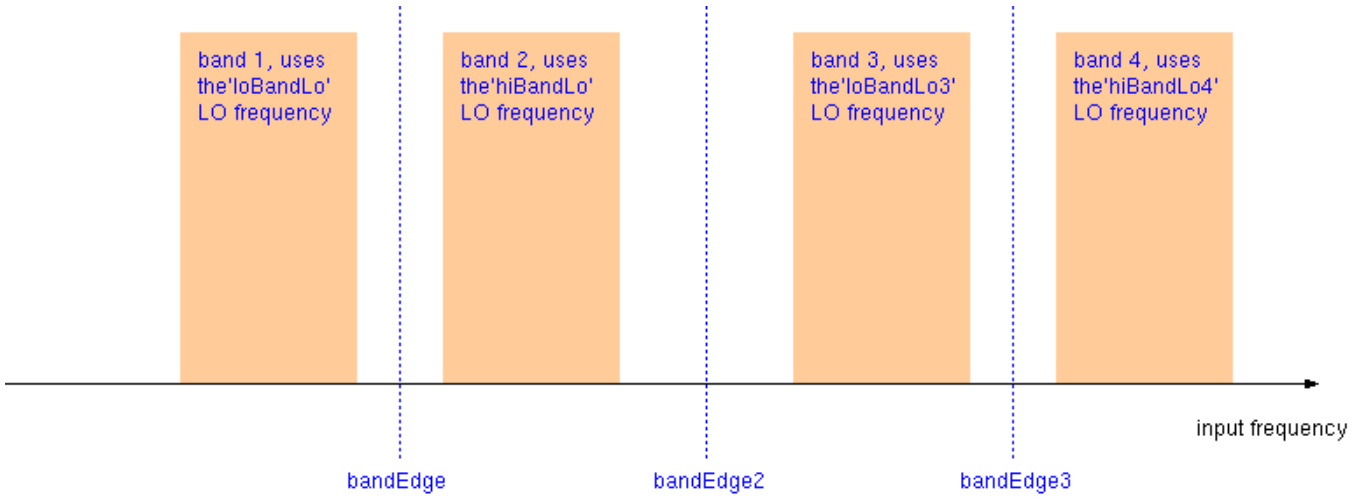
LO Frequency Calculations

The beacon receiver may store one or more local oscillator frequencies which are used to calculate the L-band frequency at the receiver input from the receive frequency at the antenna. Most beacon receivers support two frequency bands with separate LO frequencies. A model with four frequency bandy and 8-way input switch is also available. With the LO frequencies and frequency band edges setup correctly, the beacon receiver always shows and accepts the input frequency in terms of true receive frequency at the antenna rather than in terms of L-band frequency. Optionally a 22kHz tone may be modulated on the LNB power supply to switch the band.

All LO frequencies are to be specified in MHz. The LO is assumed to be below the input frequency unless the frequency is given as a negative value. In the latter case, the beacon receiver calculates for a LO frequency above the carrier.

Two-band receivers select the frequency band by the 'bandEdge' configuration parameter. Input frequencies below this value get the 'loBandLo' LO frequency applies, frequencies above the edge are calculated to L-band using the 'hiBandLo' LO frequency.

Four-band receivers provide three frequency band edges which divide the input frequency range into four bands. Four separate LO frequencies are applied are applied according to the actual band the selected frequency is within. For compatibility reasons, the lower frequency bands and the band edge in between are controlled by the parameters mentioned above. Two additional band edges and LO frequencies control the upper two frequency bands:



With the 22kHzTone parameter set to AUTO, the 22 kHz tone gets activated if the input frequency is in frequency band 2 or in in band 4.

C/N Measurement Mode

In addition to the normal level measurement, the sat-nms LBRX beacon receiver is able to perform C/N measurements. This is done by measuring the level of the received noise at a frequency other than the receive frequency in regular intervals. Each measured signal level

then gets converted to a C/N value by referring it to the noise level value. While the receiver measures the noise level, it freezes the displayed C/N value as well as the analog output voltage for this time.

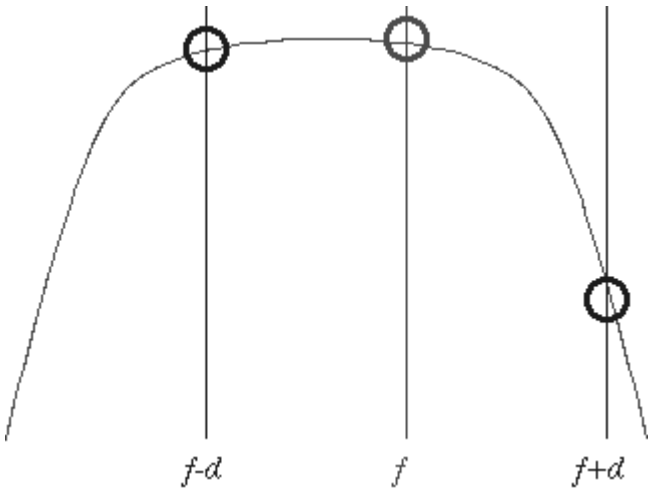
The behavior of the C/N measurement function is controlled by three operational parameters:

- *Parameter Name --- Description*
- C/N Noise measurement --- This parameter controls if the receiver shall perform a plain input level measurement or a C/N measurement. The following measurement modes are available:
  - **OFF** --- The receiver performs a plain level measurement.
  - **IN** --- The receiver measures the signal / noise ratio. The The C/N measurement page shows the C/N value in dB. and the normalizes C/No value as well.
- Noise Measurement Frequency --- This parameter specifies the frequency at which the receiver shall measure the noise level at a certain interval. Like with the receive frequency, the LO frequency settings made at the Setup page are taken into account also for this frequency value.To get reasonable results with a C/N measurement, you should consider the following:
  1. The receiver does not change the LNB frequency band setting when it switches from the level measurement to the noise measurement. The LNB probably would change it's gain in this case. The noise measurement frequency hence must be in the same frequency band as the receive frequency.
  2. Measuring the noise level at the band edge may falsify the result due to the LNB's band filter. The measured noise level may be too low in this case.
  3. You should verify with a spectrum analyzer, that no signal disturbs the noise measurement at the selected frequency.
- Noise Measurement Interval --- This parameter defines the interval at which the receiver inserts noise measurements in the C/N modes. The time is specified in seconds. 3600 secs being one hour is a suitable setting in most cases.

Frequency Tracking

The sat-nms LBRX beacon receiver using it's non-coherent receiver design, does not automatically follow an input signal drifting in frequency like a PLL receiver would do. To let the receiver compensate effects like an LNC frequency drift due to temperature or Doppler frequency shift, a frequency tracking algorithm has been implemented in the receiver.

With frequency tracking enabled, the receiver tunes the receive frequency a small amount up and down at certain intervals. Such a search step takes less than one second, while the frequency is detuned, the level reading and the analog output of the receiver stays frozen.



The step size used for a frequency search step depends in the measurement bandwidth, the receiver automatically selects the appropriate step size. Receiving a C/W signal, the receiver software expects a signal degradation of 1..3 dB when the frequency has been tuned up or down. The receiver evaluates the signal center frequency by comparing the measured level, at the old frequency *f* , at *f+d* and *f-d* to the known shape of the filter.

The behavior of the frequency track facility is controlled by three operational parameters:

Parameter Name	***Description***
Frequency Tracking	This parameter switches the the frequency tracking facility of the receiver ON or OFF. Please note, that switching off the frequency tracking does not reset the frequency offset which is applied by the frequency tracking. To reset the offset, set the receiver's frequency setting.
Frequency Tracking Interval	This parameter sets the interval on which the frequency tracking procedure operates. The value is in seconds. Recommended settings are 15 seconds to tune the receiver quickly to a frequency you do not know precisely. For normal operation a frequency tracking interval of one hour (3600 secs) is recommended.
Frequency Tracking width	With this setting you limit the frequency offset the frequency tracking procedure may apply to the nominal frequency. The frequency tracking never tunes the receiver to a frequency outside the specified range, a frequency track fault is generated if the tracked frequency reaches the limit.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

- *parameter --- description*

- address --- The IP address of the receiver to control.
- loBandLo --- Enter the LO frequency for the lower frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. 'loBandLo' is applied for frequencies below 'bandEdge'. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- hiBandLo --- Enter the LO frequency for the upper frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. 'hiBandLo' is applied for frequencies above 'bandEdge'. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- loBandLo3 --- This LO frequency ist only available with beacon receivers which support 4 frequency bands. Enter the LO frequency for the frequency band between 'bandEdge2' and 'bandEdge3' here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- hiBandLo4 --- This LO frequency ist only available with beacon receivers which support 4 frequency bands. Enter the LO frequency for the frequency band above 'bandEdge3' here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- bandEdge --- Enter the edge between the receive frequency bands here (MHz). The receiver applies the 'loBandLo' LO frequency if the receive frequency is below the edge, the 'hiBandLo' LO frequency if it is above. If the 22kHzTone parameter is set to AUTO, the 22 kHz ist set accordingly.
- bandEdge2 --- This band edge value is only available with beacon receivers which support four frequency bands. Enter the edge between the receive frequency bands 2 and 3 here (MHz). The receiver applies the 'loBandLo' and 'hiBandLo' LO frequencies if the receive frequency is below this edge, the 'loBandLo3' and 'hiBandLo4' LO frequencies if it is above.
- bandEdge3 --- This band edge value is only available with beacon receivers which support four frequency bands. Enter the edge between the upper receive frequency bands 3 and 4 here (MHz). The receiver applies the 'loBandLo3' LO frequency if the receive frequency is below the edge, the 'hiBandLo4' LO frequency if it is above. If the 22kHzTone parameter is set to AUTO, the 22 kHz ist set accordingly.
- lnbVolt --- This parameter controls the LNB supply voltage provided by the receiver at it's input connector. The following settings are available:
    - **OFF** --- The D/C voltage is completely switched off.
    - **14V** --- The LNB supply voltage is 14V
    - **18V** --- The LNB supply voltage is 18V
    - **AUTO** --- The LNB supply voltage is switched on, the voltage depends on the 'Polarization' parameter. The voltage is 14V for vertical polarization, 18V for

horizontal polarization.

- 22kHzTone --- This parameter controls the presence of a 22 kHz tone on the LNB supply voltage. The following settings are available:
  - **OFF** --- The 22 kHz tone is switched OFF.
  - **ON** --- The 22 kHz tone is switched ON.
  - **AUTO** --- The receiver automatically enables the 22 kHz tone depending on the receive frequency set. The tone is switched OFF for frequencies below the band edge, switched ON for frequencies above the band edge. The band edge is set with the 'bandEdge' parameter above.
- vOutScale --- This parameter defines the slope of the receiver's voltage output in V/dB. The output voltage has a range of 0 .. 10 V. Setting this parameter to 0.25V/dB lets the analog output cover a dynamic range of 40 dB.
- vOutOffset --- This parameter defines, which input level gives 0V output.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.note	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.loMode	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.showVoltage	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.loMode	CHOICE	SETUP	EnumRange (LEGACY 8-LO-MODE)
config.useMSwitch	CHOICE	SETUP	EnumRange (OFF ON)
config.minFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.maxFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.aliasPolH	TEXT	SAVE SETUP	StringRange
config.aliasPolV	TEXT	SAVE SETUP	StringRange
config.vOutScale	FLOAT	SETUP	DoubleRange (-5.0000 .. 5.0000)
config.vOutOffs	FLOAT	SETUP	DoubleRange (-200.00 .. 0.00)
config.lnbVolt	CHOICE	SETUP	EnumRange (OFF 14V 18V AUTO)
config.22kHzTone	CHOICE	SETUP	EnumRange (OFF ON AUTO)
config.inputs	CHOICE	SAVE SETUP	EnumRange (4 8 16)
config.commandDelay	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 1.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL-ACCESS READONLY)
config.bandEdge	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge2	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge3	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge4	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge5	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge6	FLOAT		DoubleRange (0.000 .. 25000.000)
config.bandEdge7	FLOAT		DoubleRange (0.000 .. 25000.000)

name	*type*	*flags*	*range*
config.lo1Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo2Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo3Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo4Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo5Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo6Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo7Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo8Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo9Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo10Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo11Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo12Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo13Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo14Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo15Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo16Freq	FLOAT		DoubleRange (-40000.000 .. 40000.000)
config.lo1HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)

name	*type*	*flags*	*range*
config.lo2HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo3HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo4HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo5HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo6HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo7HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo8HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo9HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo10HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo11HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo12HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo13HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo14HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo15HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo16HiLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo1LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo2LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo3LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo4LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo5LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo6LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo7LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo8LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo9LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo10LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo11LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)

name	*type*	*flags*	*range*
config.lo12LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo13LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo14LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo15LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo16LoLimit	FLOAT		DoubleRange (0.000 .. 25000.000)
config.lo1Pol	CHOICE		EnumRange (H V H/V)
config.lo2Pol	CHOICE		EnumRange (H V H/V)
config.lo3Pol	CHOICE		EnumRange (H V H/V)
config.lo4Pol	CHOICE		EnumRange (H V H/V)
config.lo5Pol	CHOICE		EnumRange (H V H/V)
config.lo6Pol	CHOICE		EnumRange (H V H/V)
config.lo7Pol	CHOICE		EnumRange (H V H/V)
config.lo8Pol	CHOICE		EnumRange (H V H/V)
config.lo9Pol	CHOICE		EnumRange (H V H/V)
config.lo10Pol	CHOICE		EnumRange (H V H/V)
config.lo11Pol	CHOICE		EnumRange (H V H/V)
config.lo12Pol	CHOICE		EnumRange (H V H/V)
config.lo13Pol	CHOICE		EnumRange (H V H/V)
config.lo14Pol	CHOICE		EnumRange (H V H/V)
config.lo15Pol	CHOICE		EnumRange (H V H/V)
config.lo16Pol	CHOICE		EnumRange (H V H/V)
config.lo1Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo2Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)

name	*type*	*flags*	*range*
config.lo3Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo4Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo5Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo6Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo7Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo8Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo9Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo10Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo11Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo12Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo13Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo14Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)

name	*type*	*flags*	*range*
config.lo15Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo16Switch	CHOICE		EnumRange (J8.1 J8.2 J8.3 J8.4 J9.1 J9.2 J9.3 J9.4 J10.1 J10.2 J10.3 J10.4 J11.1 J11.2 J11.3 J11.4)
config.lo1LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo2LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo3LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo4LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo5LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo6LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo7LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo8LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo9LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo10LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo11LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo12LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo13LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo14LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo15LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo16LnbVolt	CHOICE		EnumRange (14V 18V)
config.lo1Tone	CHOICE		EnumRange (OFF ON)
config.lo2Tone	CHOICE		EnumRange (OFF ON)
config.lo3Tone	CHOICE		EnumRange (OFF ON)
config.lo4Tone	CHOICE		EnumRange (OFF ON)
config.lo5Tone	CHOICE		EnumRange (OFF ON)
config.lo6Tone	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.lo7Tone	CHOICE		EnumRange (OFF ON)
config.lo8Tone	CHOICE		EnumRange (OFF ON)
config.lo9Tone	CHOICE		EnumRange (OFF ON)
config.lo10Tone	CHOICE		EnumRange (OFF ON)
config.lo11Tone	CHOICE		EnumRange (OFF ON)
config.lo12Tone	CHOICE		EnumRange (OFF ON)
config.lo13Tone	CHOICE		EnumRange (OFF ON)
config.lo14Tone	CHOICE		EnumRange (OFF ON)
config.lo15Tone	CHOICE		EnumRange (OFF ON)
config.lo16Tone	CHOICE		EnumRange (OFF ON)
frequency	FLOAT		DoubleRange (950.000 .. 40000.000)
attenuation	CHOICE		EnumRange (30 dB 20 dB 10 dB 0 dB)
gain	CHOICE		EnumRange (-30 dB -20 dB -10 dB 0 dB)
bandwidth	CHOICE		EnumRange (6 kHz 12 kHz 30 kHz 100 kHz)
averaging	CHOICE		EnumRange (0.1 Hz 0.2 Hz 0.5 Hz 1 Hz 2 Hz 5 Hz)
threshold	FLOAT		DoubleRange (-999.99 .. 0.00)
polarization	CHOICE		EnumRange (H V)
speccomp	CHOICE		EnumRange (OFF)
ftrack.on	CHOICE		EnumRange (OFF ON)
ftrack.width	INTEGER		IntegerRange (10 .. 1000)
ftrack.interval	INTEGER		IntegerRange (1 .. 21600)
ftrack.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ftrack.running	BOOLEAN	R/O	BooleanRange R/O (true false)
bg.inhibit	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cton	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cton0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cn.on	CHOICE		EnumRange (OFF ON)
cn.frequency	FLOAT		DoubleRange (950.000 .. 19000.000)
cn.interval	INTEGER		IntegerRange (1 .. 21600)
ssrch.active	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ssrch.enable	CHOICE		EnumRange (OFF ON)
ssrch.delay	INTEGER		IntegerRange (0 .. 600)
ssrch.force	TEXT	NOPRESET	StringRange
internal.sccl	TEXT	R/O	StringRange R/O
internal.v	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.has4lo	TEXT	R/O	StringRange R/O
internal.has8lo	TEXT	R/O	StringRange R/O
internal.haslimits	TEXT	R/O	StringRange R/O
internal.hasAlias	TEXT	R/O	StringRange R/O
internal.hasSummary	TEXT	R/O	StringRange R/O
internal.pol	CHOICE		EnumRange (H V)
internal.aliasPol	TEXT		StringRange
internal.hasNoInputs	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (D/C power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Level threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Synth lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Frequency track)

9.604 SatService-Deice

This page describes the device driver and the device window for the SatService DEICE controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands deice parameters.
- --- Shows the temperatures and output states
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- retries --- The number of communication retries which are made at application level.
- internalTempThreshold --- Internal temperature threshold
- ambientTempThreshold --- Ambient temperature threshold
- tempSlopeThreshold --- Temperature slope threshold
- measInterval --- Measurement interval

- tempHysteresis --- Temperature hysteresis
- deiceSensorDelay --- Deice sensor delay
- rainSensorDelay --- Rain sensor delay

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.internalTempThreshold	INTEGER	SETUP	IntegerRange (0 .. 45)
config.ambientTempThreshold	INTEGER	SETUP	IntegerRange (0 .. 10)
config.tempSlopeThreshold	INTEGER	SETUP	IntegerRange (0 .. 10)
config.measInterval	INTEGER	SETUP	IntegerRange (0 .. 600)
config.tempHysteresis	INTEGER	SETUP	IntegerRange (0 .. 10)
config.deiceSensorDelay	INTEGER	SETUP	IntegerRange (0 .. 59)
config.rainSensorDelay	INTEGER	SETUP	IntegerRange (0 .. 59)
inputs.0	HEX	R/O	HexRange R/O (0 .. 0)
inputs.1	HEX	R/O	HexRange R/O (0 .. 0)
inputs.2	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.01.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.02.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.03.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.04.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.05.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)

name	*type*	*flags*	*range*
output.06.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.07.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.08.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.09.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.10.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)

name	*type*	*flags*	*range*
output.11.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.12.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.13.type	CHOICE	R/O	EnumRange R/O (unused output internal blower disabled blower stage 1 blower stage 2 blower stage 3 subreflector disabled subreflector rainblower disabled rainblower)
output.01	BOOLEAN	R/O	BooleanRange R/O (true false)
output.02	BOOLEAN	R/O	BooleanRange R/O (true false)
output.03	BOOLEAN	R/O	BooleanRange R/O (true false)
output.04	BOOLEAN	R/O	BooleanRange R/O (true false)
output.05	BOOLEAN	R/O	BooleanRange R/O (true false)
output.06	BOOLEAN	R/O	BooleanRange R/O (true false)
output.07	BOOLEAN	R/O	BooleanRange R/O (true false)
output.08	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
output.09	BOOLEAN	R/O	BooleanRange R/O (true false)
output.10	BOOLEAN	R/O	BooleanRange R/O (true false)
output.11	BOOLEAN	R/O	BooleanRange R/O (true false)
output.12	BOOLEAN	R/O	BooleanRange R/O (true false)
output.13	BOOLEAN	R/O	BooleanRange R/O (true false)
output.01.name	TEXT	R/O	StringRange R/O
output.02.name	TEXT	R/O	StringRange R/O
output.03.name	TEXT	R/O	StringRange R/O
output.04.name	TEXT	R/O	StringRange R/O
output.05.name	TEXT	R/O	StringRange R/O
output.06.name	TEXT	R/O	StringRange R/O
output.07.name	TEXT	R/O	StringRange R/O
output.08.name	TEXT	R/O	StringRange R/O
output.09.name	TEXT	R/O	StringRange R/O
output.10.name	TEXT	R/O	StringRange R/O
output.11.name	TEXT	R/O	StringRange R/O
output.12.name	TEXT	R/O	StringRange R/O
output.13.name	TEXT	R/O	StringRange R/O
temperature.0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
deice.mode	CHOICE		EnumRange (OFF MANUAL AUTO)
deice.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
deice.stages	INTEGER	R/O	IntegerRange R/O (0 .. 0)
deice.remaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rainblower.mode	CHOICE		EnumRange (OFF MANUAL AUTO)
rainblower.requested	BOOLEAN	R/O	BooleanRange R/O (true false)
rainblower.remaining	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
state.sensor.1	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sensor.2	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sensor.3	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sensor.4	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sensor.5	BOOLEAN	R/O	BooleanRange R/O (true false)
state.sensor.6	BOOLEAN	R/O	BooleanRange R/O (true false)
initialized	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Output 1 Alarm 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Output 1 Alarm 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output 2 Alarm 1)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 2 Alarm 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 3 Alarm 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 3 Alarm 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 4 Alarm 1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 4 Alarm 2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output 5 Alarm 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 5 Alarm 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 6 Alarm 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 6 Alarm 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 7 Alarm 1)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 7 Alarm 2)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 8 Alarm 1)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 8 Alarm 2)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Output 9 Alarm 1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Output 9 Alarm 2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Output 10 Alarm 1)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Output 10 Alarm 2)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Output 11 Alarm 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Output 11 Alarm 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Subreflector Alarm 1)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Subreflector Alarm 2)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Rainblower Alarm 1)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Rainblower Alarm 2)

9.605 SatService-IO-FEP-Protection-1-To-1

The SatService-IO-FEP-Protection-1-To-1 device controls one 1:1 protection switching unit in a SatService IO-FEP. The driver uses SNMP for the communication to the IO-FEP, multiple instances of the device may be configured to access multiple protection switching units in the same device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the device talk to the protection switching unit it shall control.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- switchNo --- selects the protection switch unit within the IO-FEP. Actually the IO-FEP supports only on protection switch unit.
- operationMode --- may be set to SW-ONCE or SW-ALWAYS to control how the device handles consecutive faults.
- chainAssign --- by default the sat-nms driver for the IO-FEP protection switch maps the position (chain number) "1" to the switch position "A". This may be reverted by setting this configuration parameter to "1-B". This does not change anything in the IO-FEP itself, only the position indication at the sat-nms software changes.
- suppressFaultIfOff --- may set to YES or NO. If set to YES, single fault and double fault are suppressed if protection is set to OFF. If set to NO, these faults are reported even if protection is OFF.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.switchNo	INTEGER	SAVE SETUP	IntegerRange (1 .. 16)
config.operationMode	CHOICE	SETUP	EnumRange (SW-ALWAYS SW-ONCE)
config.chainAssign	CHOICE	SAVE SETUP	EnumRange (1-A 1-B)
config.suppressFaultIfOff	CHOICE	SAVE SETUP	EnumRange (YES NO)
internal.prsw	HEX	R/O	HexRange R/O (0 .. 0)
internal.posb	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.swalways	BOOLEAN	R/O	BooleanRange R/O (true false)
location	CHOICE		EnumRange (1 2)
protection	CHOICE		EnumRange (OFF ON)
reset	TEXT		StringRange
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
singleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
doubleFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switchFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switched	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Single fault)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Double fault)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switch fault)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Has switched)

9.606 SatService-IO-FEP-Protection-2-To-1

The SatService-IO-FEP-Protection-2-To-1 device controls one 2:1 protection switching unit in a SatService IO-FEP. The driver uses SNMP for the communication to the IO-FEP, multiple instances of the device may be configured to access multiple protection switching units in the same device.

The IO-FEP 2:1 redundancy circuit uses two 1:1 redundancy units linked together. Hence up to four 2:1 redundancy units (numbered 1, 3, 5, 7) are available with a basic IO-FEP, the extended IO-FEP version supports up to eight 2:1 redundancy units.

Switch positions are immutably defined by the IO-FEP: "A" means "operative chain in use", "B" means redundant chain switched in place. Based on this definition, the device translates the switch positions as follows:

- **A : A** --- Location setting "OFF", both operative chains are in use.
- **B : A** --- Location setting "1", chain 1 is replaced by the redundant chain.
- **A : B** --- Location setting "2", chain 2 is replaced by the redundant chain.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page controls the operational parameters of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the device talk to the protection switching unit it shall control.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- switchNo --- selects the protection switch unit within the IO-FEP (one of 1, 3, 5, 7, with an IO-FEP-E also 9, 11, 13 and 15)

Remarks

- Clicking "Reset" at the maintenance page clears a pending "switched" condition and switches the location back to "OFF" (switch positions A/A).
- Changing the location setting also resets a pending "switched" condition.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.switchNo	CHOICE	SAVE SETUP	EnumRange (1 3 5 7 9 11 13 15)
internal.prsw	HEX	R/O	HexRange R/O (0 .. 0)
internal.pos1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pos2	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.label	TEXT	R/O	StringRange R/O
internal.s1	TEXT	R/O	StringRange R/O
internal.s2	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (OFF 1 2)
protection	CHOICE		EnumRange (OFF ON)
chain1Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain2Fault	BOOLEAN	R/O	BooleanRange R/O (true false)
chainSFault	BOOLEAN	R/O	BooleanRange R/O (true false)
switched1	BOOLEAN	R/O	BooleanRange R/O (true false)
switched2	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Chain 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Chain 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Spare)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Chain 1 switched)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Chain 2 switched)

9.607 SatService-IO-FEP

This page describes the device driver and the device window for the *SatService-IO-FEP* device. This device monitors and controls the SatService IO-FEP controller.

The *SatService-IO-FEP* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the *SatService-IO-FEP* device driver directly (except to access the temperature readings). There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *SatService-IO-FEP* device driver as a low level driver to access the physical I/O ports.

Input ports

The variables 'inputs.0', 'inputs.1' and 'inputs.2' refer to the input ports 1..16, 17..32 and 33..48 respectively. The least significant bit in a variable reports the state of the lowest numbered port in it's group.

The variables report the logical state if the input. A bit set to '1' denotes an 'ON' or 'FLT' state, '0' stands for 'OFF' or 'OK'. The reported input states respect the 'delay' and 'invert' settings configured in the IO-FEP for each individual port.

Unused ports always report a '0', the same applies to the input ports 33..48 if the IO-FEP has no extension board installed.

Output ports

The variable 'outputs' reports the actual state of the digital outputs and also sets the output states. The least significant bit in the variable corresponds to output 1, the most significant bit to output 16. The port states respect the 'invert' setting configured for each individual port in the IO-FEP.

Unused ports and ports which are configured in the IO-FEP as HPA muting outputs always report the state '0'. Writing to these ports will be ignored.

Waveguide switches

The variable 'wgstat' reports the actual state of the waveguide switches. For each waveguide switch there are 2 bits, indicating the position of the switch. '01' denotes position 'A', '10' denotes position 'B'. The bit combinations '00' and '11' both indicate a fault. This similar to the way, the physical position indicators of a waveguide switch are working. However, the bits reflect a logical state in the IO-FEP rather than the physical state of the position indicators. This is because the IO-FEP hides it's internal queuing mechanism for switch actuation's from the controlling M&C.

- bit 0 --- 0x00000001switch 1, position A
- bit 1 --- 0x00000002switch 1, position B
- bit 2 --- 0x00000004switch 2, position A
- bit 3 --- 0x00000008switch 2, position B
- bit 4 --- 0x00000010switch 3, position A
- bit 5 --- 0x00000020switch 3, position B
- bit 6 --- 0x00000040switch 4, position A
- bit 7 --- 0x00000080switch 4, position B
- bit 8 --- 0x00000100switch 5, position A
- bit 9 --- 0x00000200switch 5, position B
- bit 10 --- 0x00000400switch 6, position A
- bit 11 --- 0x00000800switch 6, position B
- bit 12 --- 0x00001000switch 7, position A
- bit 13 --- 0x00002000switch 7, position B
- bit 14 --- 0x00004000switch 8, position A
- bit 15 --- 0x00008000switch 8, position B

- bit 16 --- 0x00010000switch 9, position A
- bit 17 --- 0x00020000switch 9, position B
- bit 18 --- 0x00040000switch 10, position A
- bit 19 --- 0x00080000switch 10, position B
- bit 20 --- 0x00100000switch 11, position A
- bit 21 --- 0x00200000switch 11, position B
- bit 22 --- 0x00400000switch 12, position A
- bit 23 --- 0x00800000switch 12, position B
- bit 24 --- 0x01000000switch 13, position A
- bit 25 --- 0x02000000switch 13, position B
- bit 26 --- 0x04000000switch 14, position A
- bit 27 --- 0x08000000switch 14, position B
- bit 28 --- 0x10000000switch 15, position A
- bit 29 --- 0x20000000switch 15, position B
- bit 30 --- 0x40000000switch 16, position A
- bit 31 --- 0x80000000switch 16, position B

The variable 'wgcmd' is the complement to 'wgstat' for actuating the switches. In the software the bits of this variable are treated like output ports which perform the switch motor actuation. In fact, the IO-FEP monitors 0 to 1 transitions of each bit and performs the switch actuation accordingly. The actuation pulse duration is controlled by the IO-FEP, not by the sat-nms software. Nevertheless, the IO-FEP waveguide switch interface is fully compatible to the WG-Switch logical device.

Switches configured in the IO-FEP as 'unused' always report the position '00'. The same applies to the switches 9..16 if no extension board is installed in the IO-FEP.

PSU fault monitoring

The IO-FEP "E" version, together with firmware versions beginning from January 2013, monitors the presence of power supply voltages. The sat-nms device driver reports these PSU fault flags as 'faults.11' to 'faults.14'. With IO-FEP versions that do not support PSU monitoring, these fault flags remain unset / uninitialized.

The PSU monitoring for the redundant supply voltages may be masked in the IO-FEPs setup. The IO-FEP reports always them as OK in this case.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the SatService IO-FEP acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the IO-FEP device in the device setup window , there must not be defined any other devices beside the IO-FEP for this interface.
- The IDLE time for the interface must be set to 200 msecs
- In the IO-FEP's device setup dialog, the 'retries' parameters must be set to '0', the snmpTimeout to '200 msec' and the snmpRetries parameter to '2'

- Any WG-Switch configured to work via the IO-FEP must use a pulseWidth of 1500 msecs or longer.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- Shows the temperature readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- retries --- The number of communication retries which are made at application level.**For use with the sat-nms framework for M&C of plain digital I/O functions this parameter must be set to 0.**
- snmpTimeout --- The timeout value used by the SNMP protocol implementation. The default value is 5 seconds. **For use with the sat-nms framework for M&C of plain digital I/O functions this parameter must be set to 200 msec.**
- snmpRetries --- The the number of retries used by the SNMP protocol implementation. The default value is 2 retries.

- swCtrlFaultSrc --- Defines in which way the driver evaluates the switch control fault flag (faults.17). This fault flag tells that the IO-FEP actually is not capable to control switches or to report input signals correctly.
    - COMM --- The switch control fault is determined only from the communication fault flag, the PSU fault flags are ignored. This settings should be used if IO-FEP (not the extended version) is controlled
    - COMM+PSU1 --- The switch control fault is set if the IO-FEP either shows a communication fault or the primary EXT24 voltage fails. This setting should be used if controlling an IO-FEP-E with a single EXT24 power supply.
    - COMM+PSU1+PSU2 --- The switch control fault is set if the IO-FEP either shows a communication fault or both EXT24 voltages fails. This setting should be used if controlling an IO-FEP-E with redundant EXT24 power supplies.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.extensionBoard	TEXT	R/O	StringRange R/O
info.psuMonitoring	BOOLEAN	R/O	BooleanRange R/O (true false)
info.ntpSyncSupport	BOOLEAN	R/O	BooleanRange R/O (true false)
snmpTimeout	INTEGER	SETUP	IntegerRange (1 .. 100000)
snmpRetries	INTEGER	SETUP	IntegerRange (0 .. 10)
config.swCtrlFaultSrc	CHOICE	SAVE SETUP	EnumRange (COMM COMM+PSU1 COMM+PSU1+PSU2)
internal.caps	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.dummy	TEXT	R/O	StringRange R/O
internal.ltmp	HEX	R/O	HexRange R/O (0 .. 0)
internal.htmp	HEX	R/O	HexRange R/O (0 .. 0)
internal.ntp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
inputs.0	HEX	R/O	HexRange R/O (0 .. 0)
inputs.1	HEX	R/O	HexRange R/O (0 .. 0)
inputs.2	HEX	R/O	HexRange R/O (0 .. 0)
outputs	HEX		HexRange (0 .. 0)
wgstat	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd	HEX		HexRange (0 .. 0)
temperature.0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
initialized	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Board temperature LOW)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Board temperature HIGH)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature 1 LOW)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature 1 HIGH)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature 2 LOW)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature 2 HIGH)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature 3 LOW)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature 3 HIGH)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Temperature 4 LOW)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Temperature 4 HIGH)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU 24V DC)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU 24V EXT)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU 24V DC RED.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 24V EXT RED.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (NTP sync)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Primary NTP server)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch control)

9.608 SatService-IO-FEP2

This page describes the device driver and the device window for the*SatService-IO-FEP2* device. This device monitors and controls the SatService IO-FEP controller.

The *SatService-IO-FEP2* device driver is part of the sat-nms framework for M&C of plain digital I/O functions. Monitoring alarm circuits or controlling waveguide switches are indicatives for applications of this framework.

The application configuration normally does not access the*SatService-IO-FEP2* device driver directly (except to access the temperature or LNA monitoring readings). There are a couple of application level logical devices (e.g. Input , Output or WG-Switch) which may use the *SatService-IO-FEP* device driver as a low level driver to access the physical I/O ports.

Input ports

The variables 'inputs.0', 'inputs.1' and 'inputs.2' refer to the input ports 1..16, 17..32 and 33..48 respectively. The least significant bit in a variable reports the state of the lowest numbered port in it's group.

The variables report the logical state if the input. A bit set to '1' denotes an 'ON' or 'FLT' state,

'0' stands for 'OFF' or 'OK'. The reported input states respect the 'delay' and 'invert' settings configured in the IO-FEP for each individual port.

Unused ports always report a '0', the same applies to input ports which are not licensed for this IO-FEP.

Output ports

The variable 'outputs' reports the actual state of the digital outputs and also sets the output states. The least significant bit in the variable corresponds to output 1, the most significant bit to output 16. The port states respect the 'invert' setting configured for each individual port in the IO-FEP.

Unused ports and ports which are configured in the IO-FEP as HPA muting outputs always report the state '0'. Writing to these ports will be ignored.

Beside the 'outputs' variable the IO-FEP provides 16 variables 'output.01' ... 'output.16', each reporting / commanding the state of one of the IO-FEP's outputs as 0/1. New applications should use these individual output variables instead of the common 'outputs' bitmask.

Waveguide switches

The variable 'wgstat' reports the actual state of the waveguide switches. For each waveguide switch there are 2 bits, indicating the position of the switch. '01' denotes position 'A', '10' denotes position 'B'. The bit combinations '00' and '11' both indicate a fault. This similar to the way, the physical position indicators of a waveguide switch are working. However, the bits reflect a logical state in the IO-FEP rather than the physical state of the position indicators. This is because the IO-FEP hides it's internal queuing mechanism for switch actuation's from the controlling M&C.

bit	bit mask	WG switch	position
bit 0	0x00000001	switch 1	position A
bit 1	0x00000002	switch 1	position B
bit 2	0x00000004	switch 2	position A
bit 3	0x00000008	switch 2	position B
bit 4	0x00000010	switch 3	position A
bit 5	0x00000020	switch 3	position B
bit 6	0x00000040	switch 4	position A
bit 7	0x00000080	switch 4	position B
bit 8	0x00000100	switch 5	position A
bit 9	0x00000200	switch 5	position B

bit	bit mask	WG switch	position
bit 10	0x00000400	switch 6	position A
bit 11	0x00000800	switch 6	position B
bit 12	0x00001000	switch 7	position A
bit 13	0x00002000	switch 7	position B
bit 14	0x00004000	switch 8	position A
bit 15	0x00008000	switch 8	position B
bit 16	0x00010000	switch 9	position A
bit 17	0x00020000	switch 9	position B
bit 18	0x00040000	switch 10	position A
bit 19	0x00080000	switch 10	position B
bit 20	0x00100000	switch 11	position A
bit 21	0x00200000	switch 11	position B
bit 22	0x00400000	switch 12	position A
bit 23	0x00800000	switch 12	position B
bit 24	0x01000000	switch 13	position A
bit 25	0x02000000	switch 13	position B
bit 26	0x04000000	switch 14	position A
bit 27	0x08000000	switch 14	position B
bit 28	0x10000000	switch 15	position A
bit 29	0x20000000	switch 15	position B
bit 30	0x40000000	switch 16	position A
bit 31	0x80000000	switch 16	position B

The variable 'wgcmd' is the complement to 'wgstat' for actuating the switches. In the software the bits of this variable are treated like output ports which perform the switch motor actuation. In fact, the IO-FEP monitors 0 to 1 transitions of each bit and performs the switch actuation accordingly. The actuation pulse duration is controlled by the IO-FEP, not by the sat-nms software. Nevertheless, the IO-FEP waveguide switch interface is fully compatible to the WG-Switch logical device.

Switches configured in the IO-FEP as 'unused' always report the position '00'. The same applies to switches which are not licensed for this IO-FEP.

Configuration settings

If used with the sat-nms framework for M&C of plain digital I/O functions, the SatService IO-FEP acts like an unprogrammed PLC device. It is intended, that the device driver processes switch settings and changes of input circuits as fast as possible. The following settings are necessary to achieve this:

- When defining the IO-FEP device in the device setup window, there must not be defined any other devices beside the IO-FEP for this interface.
- The *SatService-IO-FEP2* device driver uses the Modbus/TCP protocol to communicate with the IO-FEP. The IO-FEP's IP address is configured in the device setup window as the interface for the device. It uses port number 502 (e.g. 10.0.0.1:502), the Modbus/TCP standard port number.
- The IDLE time for the interface must be set to 100-200 milliseconds.
- Any WG-Switch configured to work via the IO-FEP must use a pulseWidth of 1500 msecs or longer.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands I/O port states.
- --- Shows the temperature readings
- --- Shows the LNA monitoring readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- Should be left empty
- swCtrlFaultSrc --- Defines in which way the driver evaluates the switch control fault flag (faults.17). This fault flag tells that the IO-FEP actually is not capable to control switches or to report input signals correctly.
  - COMM --- The switch control fault is determined only from the communication fault flag, the PSU fault flags are ignored.
  - COMM+PSU1 --- The switch control fault is set if the IO-FEP either shows a communication fault or the primary EXT24 voltage fails.

- COMM+PSU1+PSU2 --- The switch control fault is set if the IO-FEP either shows a communication fault or both EXT24 voltages fails.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.lnaMonitoring	BOOLEAN	R/O	BooleanRange R/O (true false)
config.swCtrlFaultSrc	CHOICE	SAVE SETUP	EnumRange (COMM COMM+PSU1 COMM+PSU1+PSU2)
config.temp.loThreshold.0	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.hiThreshold.0	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.loThreshold.1	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.hiThreshold.1	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.loThreshold.2	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.hiThreshold.2	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)

name	*type*	*flags*	*range*
config.temp.loThreshold.3	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.hiThreshold.3	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.loThreshold.4	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.temp.hiThreshold.4	FLOAT	SETUP	DoubleRange (-100.0 .. 100.0)
config.voltage.loThreshold.1	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.voltage.loThreshold.2	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.voltage.loThreshold.3	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.voltage.loThreshold.4	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.voltage.loThreshold.5	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.voltage.loThreshold.6	FLOAT	SETUP	DoubleRange (0.0 .. 30.0)
config.current.loThreshold.1	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.1	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.loThreshold.2	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.2	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.loThreshold.3	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.3	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.loThreshold.4	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.4	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.loThreshold.5	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.5	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.loThreshold.6	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
config.current.hiThreshold.6	FLOAT	SETUP	DoubleRange (0.0 .. 1000.0)
i.0	HEX	R/O	HexRange R/O (0 .. 0)
i.1	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
i.2	HEX	R/O	HexRange R/O (0 .. 0)
w.0	HEX	R/O	HexRange R/O (0 .. 0)
w.1	HEX	R/O	HexRange R/O (0 .. 0)
internal.dummy	TEXT	R/O	StringRange R/O
inputs.0	HEX	R/O	HexRange R/O (0 .. 0)
inputs.1	HEX	R/O	HexRange R/O (0 .. 0)
inputs.2	HEX	R/O	HexRange R/O (0 .. 0)
outputs	HEX	NOCOMPARE	HexRange (0 .. 0)
output.01	HEX	NOCOMPARE	HexRange (0 .. 0)
output.02	HEX	NOCOMPARE	HexRange (0 .. 0)
output.03	HEX	NOCOMPARE	HexRange (0 .. 0)
output.04	HEX	NOCOMPARE	HexRange (0 .. 0)
output.05	HEX	NOCOMPARE	HexRange (0 .. 0)
output.06	HEX	NOCOMPARE	HexRange (0 .. 0)
output.07	HEX	NOCOMPARE	HexRange (0 .. 0)
output.08	HEX	NOCOMPARE	HexRange (0 .. 0)
output.09	HEX	NOCOMPARE	HexRange (0 .. 0)
output.10	HEX	NOCOMPARE	HexRange (0 .. 0)
output.11	HEX	NOCOMPARE	HexRange (0 .. 0)
output.12	HEX	NOCOMPARE	HexRange (0 .. 0)
output.13	HEX	NOCOMPARE	HexRange (0 .. 0)
output.14	HEX	NOCOMPARE	HexRange (0 .. 0)
output.15	HEX	NOCOMPARE	HexRange (0 .. 0)
output.16	HEX	NOCOMPARE	HexRange (0 .. 0)
wgstat	HEX	R/O	HexRange R/O (0 .. 0)
wgcmd	HEX	NOCOMPARE	HexRange (0 .. 0)

name	*type*	*flags*	*range*
temperature.0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.5	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.6	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
current.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current.4	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
current.6	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
lnastat	HEX	R/O	HexRange R/O (0 .. 0)
initialized	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Board temperature LOW)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Board temperature HIGH)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Board temperature SUM)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature 1 LOW)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature 1 HIGH)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Temperature 1 SUM)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temperature 2 LOW)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature 2 HIGH)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Temperature 2 SUM)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature 3 LOW)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature 3 HIGH)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Temperature 3 SUM)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Temperature 4 LOW)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Temperature 4 HIGH)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Temperature 4 SUM)
faults.11	ALARM	R/O	AlarmFlagRange R/O (PSU 24V DC)
faults.12	ALARM	R/O	AlarmFlagRange R/O (PSU 24V EXT)
faults.13	ALARM	R/O	AlarmFlagRange R/O (PSU 24V DC RED.)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (PSU 24V EXT RED.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (NTP sync)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Primary NTP server)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Switch control)
faults.18	ALARM	R/O	AlarmFlagRange R/O (LNA PS main)
faults.19	ALARM	R/O	AlarmFlagRange R/O (LNA PS backup)
faults.41	ALARM	R/O	AlarmFlagRange R/O (LNA1 voltage low)
faults.42	ALARM	R/O	AlarmFlagRange R/O (LNA2 voltage low)
faults.43	ALARM	R/O	AlarmFlagRange R/O (LNA3 voltage low)
faults.44	ALARM	R/O	AlarmFlagRange R/O (LNA4 voltage low)
faults.45	ALARM	R/O	AlarmFlagRange R/O (LNA5 voltage low)
faults.46	ALARM	R/O	AlarmFlagRange R/O (LNA6 voltage low)
faults.51	ALARM	R/O	AlarmFlagRange R/O (LNA1 current low)
faults.52	ALARM	R/O	AlarmFlagRange R/O (LNA2 current low)
faults.53	ALARM	R/O	AlarmFlagRange R/O (LNA3 current low)
faults.54	ALARM	R/O	AlarmFlagRange R/O (LNA4 current low)

name	*type*	*flags*	*range*
faults.55	ALARM	R/O	AlarmFlagRange R/O (LNA5 current low)
faults.56	ALARM	R/O	AlarmFlagRange R/O (LNA6 current low)
faults.61	ALARM	R/O	AlarmFlagRange R/O (LNA1 current high)
faults.62	ALARM	R/O	AlarmFlagRange R/O (LNA2 current high)
faults.63	ALARM	R/O	AlarmFlagRange R/O (LNA3 current high)
faults.64	ALARM	R/O	AlarmFlagRange R/O (LNA4 current high)
faults.65	ALARM	R/O	AlarmFlagRange R/O (LNA5 current high)
faults.66	ALARM	R/O	AlarmFlagRange R/O (LNA6 current high)

9.609 SatService-LBRX-1MT

This page describes the device driver and the device window for the sat-nms LBRX-1MT beacon receiver. This version of the beacon receiver is specialized for a high measurement rate, leaving out the frequency-tracking and c/n measurement modes of the standard sat-nms beacon receiver.

The LBRX-1MT does not provide a TCP/IP interface, if gets controlled via RS232 (Miteq-MOD95 protocol). The device driver for the LBRX-1MT has been designed to take advantage of the high measurement rate of the device. The 'power' device driver variable is defines with a 1 msec update rate, making the real update cycle solely dependent on the 'IDLE' time setting defined to the receiver's interface in the VLC device setup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the receive level and lets you set the basic receive parameters like frequency, bandwidth and input attenuation.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

- *parameter --- description*
- address --- The LBRX-1MT receiver uses the fixed device address 'A'.
- loBandLo --- Enter the LO frequency for the lower frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- hiBandLo --- Enter the LO frequency for the upper frequency band here (MHz), if you want to specify the receive frequency in terms of true RF receive frequency. Enter 0 if you want the receive frequency to be displayed as receiver input (L-band) frequency.
- bandEdge --- Enter the edge between the receive frequency bands here (MHz). The receiver applies the 'loBandLo' LO frequency if the receive frequency is below the edge, the 'hiBandLo' LO frequency if it is above. If the 22kHzTone parameter is set to AUTO, the 22 kHz ist set accordingly.
- lnbVolt --- This parameter controls the LNB supply voltage provided by the receiver at it's input connector. The following settings are available:
  - **OFF** --- The D/C voltage is completely switched off.
  - **14V** --- The LNB supply voltage is 14V
  - **18V** --- The LNB supply voltage is 18V
  - **AUTO** --- The LNB supply voltage is switched on, the voltage depends on the 'Polarization' parameter. The voltage is 14V for vertical polarization, 18V for horizontal polarization.
- 22kHzTone --- This parameter controls the presence of a 22 kHz tone on the LNB supply voltage. The following settings are available:
  - **OFF** --- The 22 kHz tone is switched OFF.
  - **ON** --- The 22 kHz tone is switched ON.
  - **AUTO** --- The receiver automatically enables the 22 kHz tone depending on the receive frequency set. The tone is switched OFF for frequencies below the band edge, switched ON for frequencies above the band edge. The band edge is set with the 'bandEdge' parameter above.
- vOutScale --- This parameter defines the slope of the receiver's voltage output in V/dB. The output voltage has a range of 0 .. 10 V. Setting this parameter to 0.25V/dB lets the analog output cover a dynamic range of 40 dB.
- vOutOffset --- This parameter defines, which input level gives 0V output.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.note	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.loBandLo	FLOAT	SETUP	DoubleRange (-19000.000 .. 19000.000)
config.hiBandLo	FLOAT	SETUP	DoubleRange (-19000.000 .. 19000.000)
config.bandEdge	FLOAT	SETUP	DoubleRange (0.000 .. 19000.000)
config.lnbVolt	CHOICE	SETUP	EnumRange (OFF 14V 18V AUTO)
config.22kHzTone	CHOICE	SETUP	EnumRange (OFF ON AUTO)
config.vOutScale	FLOAT	SETUP	DoubleRange (-5.00 .. 5.00)
config.vOutOffs	FLOAT	SETUP	DoubleRange (-200.00 .. 0.00)
frequency	FLOAT		DoubleRange (950.000 .. 19000.000)
gain	CHOICE		EnumRange (-30 dB -20 dB -10 dB 0 dB)
bandwidth	CHOICE		EnumRange (6 kHz 12 kHz 30 kHz 100 kHz)
averaging	CHOICE		EnumRange (OFF 0.1 Hz 0.5 Hz 1 Hz 5 Hz 10 Hz 50 Hz 100 Hz)
threshold	FLOAT		DoubleRange (-999.99 .. 0.00)
polarization	CHOICE		EnumRange (H V)

name	*type*	*flags*	*range*
speccomp	FLOAT		DoubleRange (-10.00 .. 10.00)
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (D/C power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Level threshold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Synth lock)

9.610 SatService-LFRXTX

This page describes the device driver and the device window for the SatService L-Band Optical Links LFRX/TX. The drivers supports all configurations of the LFRX/TX chassis: Single, Master or Slave. But if the unit type changes it requires restarting of the M&C service, because the unit type is only read one time at startup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Master unit page
- --- Slave unit page
- --- Redundancy switching
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The SatService LFRX/TX units do not need any configuration parameters except the settings for the SNMP interface

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.

- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitType	CHOICE	R/O	EnumRange R/O (Master Slave Single)
info.redDefinition	TEXT	R/O	StringRange R/O
info.master.serialno	TEXT	R/O	StringRange R/O
info.master.version	TEXT	R/O	StringRange R/O
info.master.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.master	TEXT	R/O	StringRange R/O
info.slave.version	TEXT	R/O	StringRange R/O
info.slave.serialno	TEXT	R/O	StringRange R/O
info.slave.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.slave	TEXT	R/O	StringRange R/O
info.masterSlaveCom	CHOICE	R/O	EnumRange R/O (NONE HTTP RS-232)
info.correspAddress	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.masterNote	TEXT	R/O	StringRange R/O
config.slaveNote	TEXT	R/O	StringRange R/O
internal.unitType	TEXT	R/O	StringRange R/O
internal.redDefinition	TEXT	R/O	StringRange R/O
internal.master	TEXT	R/O	StringRange R/O
internal.slave	TEXT	R/O	StringRange R/O
internal.majorVer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.hasWnom	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasAttnOffset	BOOLEAN	R/O	BooleanRange R/O (true false)
master.01.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.02.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.03.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.04.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.05.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.06.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.07.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.08.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.09.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.10.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
master.01.note	TEXT	R/O	StringRange R/O
master.02.note	TEXT	R/O	StringRange R/O
master.03.note	TEXT	R/O	StringRange R/O
master.04.note	TEXT	R/O	StringRange R/O
master.05.note	TEXT	R/O	StringRange R/O
master.06.note	TEXT	R/O	StringRange R/O
master.07.note	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
master.08.note	TEXT	R/O	StringRange R/O
master.09.note	TEXT	R/O	StringRange R/O
master.10.note	TEXT	R/O	StringRange R/O
slave.01.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.02.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.03.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.04.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.05.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.06.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.07.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.08.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.09.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.10.card	CHOICE	R/O	EnumRange R/O (NONE RX TX)
slave.01.note	TEXT	R/O	StringRange R/O
slave.02.note	TEXT	R/O	StringRange R/O
slave.03.note	TEXT	R/O	StringRange R/O
slave.04.note	TEXT	R/O	StringRange R/O
slave.05.note	TEXT	R/O	StringRange R/O
slave.06.note	TEXT	R/O	StringRange R/O
slave.07.note	TEXT	R/O	StringRange R/O
slave.08.note	TEXT	R/O	StringRange R/O
slave.09.note	TEXT	R/O	StringRange R/O
slave.10.note	TEXT	R/O	StringRange R/O
protection.left	CHOICE		EnumRange DIS (manual auto off)
internal.left	CHOICE		EnumRange DIS (manual auto off)
protection.right	CHOICE		EnumRange DIS (manual auto off)
internal.right	CHOICE		EnumRange DIS (manual auto off)

name	*type*	*flags*	*range*
location.left	CHOICE		EnumRange DIS (Slot 3 Standby Slot 1 Slot 2 Slot 4 Slot 5)
location.right	CHOICE		EnumRange DIS (Slot 8 Standby Slot 6 Slot 7 Slot 9 Slot 10)
switched	INTEGER		IntegerRange DIS (0 .. 0)
protection.1	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
protection.2	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
protection.3	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
protection.4	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
protection.5	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
internal.p1	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
internal.p2	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
internal.p3	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
internal.p4	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
internal.p5	CHOICE		EnumRange DIS (OFF MANUAL AUTO)
location.1	CHOICE		EnumRange DIS (1 2)
location.2	CHOICE		EnumRange DIS (1 2)
location.3	CHOICE		EnumRange DIS (1 2)
location.4	CHOICE		EnumRange DIS (1 2)
location.5	CHOICE		EnumRange DIS (1 2)
protection.01	CHOICE		EnumRange (YES NO)
protection.02	CHOICE		EnumRange (YES NO)
protection.03	CHOICE		EnumRange (YES NO)
protection.04	CHOICE		EnumRange (YES NO)
protection.05	CHOICE		EnumRange (YES NO)
protection.06	CHOICE		EnumRange (YES NO)

name	*type*	*flags*	*range*
protection.07	CHOICE		EnumRange (YES NO)
protection.08	CHOICE		EnumRange (YES NO)
protection.09	CHOICE		EnumRange (YES NO)
protection.10	CHOICE		EnumRange (YES NO)
master.01.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.02.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.03.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.04.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.05.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.06.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.07.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.08.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.09.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.10.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.01.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.02.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.03.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.04.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.05.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.06.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.07.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.08.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.09.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
slave.10.rfThreshold	INTEGER		IntegerRange (-99 .. 0)
master.01.attenuator	INTEGER		IntegerRange (0 .. 31)
master.02.attenuator	INTEGER		IntegerRange (0 .. 31)

name	*type*	*flags*	*range*
master.03.attenuator	INTEGER		IntegerRange (0 .. 31)
master.04.attenuator	INTEGER		IntegerRange (0 .. 31)
master.05.attenuator	INTEGER		IntegerRange (0 .. 31)
master.06.attenuator	INTEGER		IntegerRange (0 .. 31)
master.07.attenuator	INTEGER		IntegerRange (0 .. 31)
master.08.attenuator	INTEGER		IntegerRange (0 .. 31)
master.09.attenuator	INTEGER		IntegerRange (0 .. 31)
master.10.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.01.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.02.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.03.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.04.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.05.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.06.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.07.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.08.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.09.attenuator	INTEGER		IntegerRange (0 .. 31)
slave.10.attenuator	INTEGER		IntegerRange (0 .. 31)
master.01.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.02.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.03.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.04.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.05.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.06.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.07.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.08.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)

name	*type*	*flags*	*range*
master.09.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.10.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.01.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.02.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.03.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.04.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.05.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.06.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.07.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.08.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.09.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
slave.10.attnOffset	INTEGER		IntegerRange DIS (-10 .. 10)
master.01.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.02.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.03.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.04.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.05.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.06.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.07.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.08.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.09.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.10.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.01.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.02.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.03.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.04.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.05.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
slave.06.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.07.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.08.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.09.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
slave.10.lnbPower	CHOICE	R/O	EnumRange R/O (OFF ON)
master.01.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.02.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.03.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.04.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.05.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.06.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.07.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.08.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.09.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.10.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.01.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.02.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.03.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.04.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.05.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.06.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.07.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.08.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.09.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.10.rfPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
master.01.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.02.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.03.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.04.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.05.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.06.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.07.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.08.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.09.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.10.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.01.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.02.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.03.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.04.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.05.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.06.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.07.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.08.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.09.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.10.diodePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.01.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.02.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.03.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.04.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.05.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.06.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.07.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
master.08.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.09.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.10.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.01.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.02.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.03.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.04.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.05.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.06.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.07.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.08.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.09.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
slave.10.optPower	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
master.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.09.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
master.10.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.01.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.02.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slave.03.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.04.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.05.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.06.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.07.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.08.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.09.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slave.10.lnbCurrent	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Master PS 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slave PS 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Master PS 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slave PS 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Master RF Power)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slave RF Power)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Master Opt.Power)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slave Opt.Power)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Master LNB Power)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slave LNB Power)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Master Threshold)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slave Threshold)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Master/Slave Comm.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slave/Master Comm.)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Internal Comm.)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Int/Extq. Switch)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Master Attenuation)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slave Attenuation)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 01 Master)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 02 Master)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 03 Master)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 04 Master)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 05 Master)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 06 Master)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 07 Master)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 08 Master)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 09 Master)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 10 Master)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 01 Slave)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 02 Slave)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 03 Slave)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 04 Slave)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 05 Slave)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 06 Slave)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 07 Slave)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 08 Slave)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 09 Slave)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Slot/Card 10 Slave)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Protection 1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Protection 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Protection 3)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Protection 4)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Protection 5)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Protection Left)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Protection Right)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Has switched)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Attn Low 01 Master)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Attn Low 02 Master)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Attn Low 03 Master)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Attn Low 04 Master)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Attn Low 05 Master)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Attn Low 06 Master)

name	*type*	*flags*	*range*
faults.57	ALARM	R/O	AlarmFlagRange R/O (Attn Low 07 Master)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Attn Low 08 Master)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Attn Low 09 Master)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Attn Low 10 Master)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Attn High 01 Master)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Attn High 02 Master)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Attn High 03 Master)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Attn High 04 Master)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Attn High 05 Master)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Attn High 06 Master)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Attn High 07 Master)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Attn High 08 Master)
faults.69	ALARM	R/O	AlarmFlagRange R/O (Attn High 09 Master)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Attn High 10 Master)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Attn Low 01 Slave)
faults.72	ALARM	R/O	AlarmFlagRange R/O (Attn Low 02 Slave)
faults.73	ALARM	R/O	AlarmFlagRange R/O (Attn Low 03 Slave)
faults.74	ALARM	R/O	AlarmFlagRange R/O (Attn Low 04 Slave)

name	*type*	*flags*	*range*
faults.75	ALARM	R/O	AlarmFlagRange R/O (Attn Low 05 Slave)
faults.76	ALARM	R/O	AlarmFlagRange R/O (Attn Low 06 Slave)
faults.77	ALARM	R/O	AlarmFlagRange R/O (Attn Low 07 Slave)
faults.78	ALARM	R/O	AlarmFlagRange R/O (Attn Low 08 Slave)
faults.79	ALARM	R/O	AlarmFlagRange R/O (Attn Low 09 Slave)
faults.80	ALARM	R/O	AlarmFlagRange R/O (Attn Low 10 Slave)
faults.81	ALARM	R/O	AlarmFlagRange R/O (Attn High 01 Slave)
faults.82	ALARM	R/O	AlarmFlagRange R/O (Attn High 02 Slave)
faults.83	ALARM	R/O	AlarmFlagRange R/O (Attn High 03 Slave)
faults.84	ALARM	R/O	AlarmFlagRange R/O (Attn High 04 Slave)
faults.85	ALARM	R/O	AlarmFlagRange R/O (Attn High 05 Slave)
faults.86	ALARM	R/O	AlarmFlagRange R/O (Attn High 06 Slave)
faults.87	ALARM	R/O	AlarmFlagRange R/O (Attn High 07 Slave)
faults.88	ALARM	R/O	AlarmFlagRange R/O (Attn High 08 Slave)
faults.89	ALARM	R/O	AlarmFlagRange R/O (Attn High 09 Slave)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Attn High 10 Slave)
master.01.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.02.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.03.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.04.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.05.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
master.06.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.07.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.08.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.09.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.10.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.01.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.02.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.03.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.04.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.05.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.06.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.07.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.08.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.09.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.10.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.01.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.02.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.03.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.04.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.05.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.06.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.07.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.08.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.09.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.10.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
master.01.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.02.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.03.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.04.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.05.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.06.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.07.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.08.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.09.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.10.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.01.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.02.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.03.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.04.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.05.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.06.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.07.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.08.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.09.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
master.10.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.01.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.02.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.03.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.04.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.05.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.06.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slave.07.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.08.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.09.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.10.rfFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.01.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.02.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.03.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.04.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.05.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.06.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.07.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.08.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.09.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.10.optFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.01.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.02.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.03.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.04.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.05.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.06.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.07.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.08.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.09.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.10.lnbFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.01.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.02.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slave.03.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.04.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.05.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.06.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.07.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.08.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.09.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.10.thrFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.01.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.02.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.03.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.04.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.05.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.06.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.07.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.08.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.09.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slave.10.lclFault	BOOLEAN	R/O	BooleanRange R/O (true false)

9.611 SatService-Matrix

This page describes the device driver and the device window for a SatService L-Band Matrix with up to 32 inputs and outputs. If the number of interfaces changes it requires restarting of the M&C service, because the the interface number is only read one time at startup.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Channel Select
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The SatService Matrix do not need any configuration parameters except the settings for the SNMP interface

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.02	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.03	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.04	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.05	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.06	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.07	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.08	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.09	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.10	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.11	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)

name	*type*	*flags*	*range*
output.12	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.13	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.14	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.15	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.16	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.17	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.18	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.19	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.20	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.21	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.22	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.23	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)

name	*type*	*flags*	*range*
output.24	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.25	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.26	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.27	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.28	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.29	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.30	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.31	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
output.32	CHOICE		EnumRange DIS (00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33)
internal.output.01	TEXT	R/O	StringRange R/O
internal.output.02	TEXT	R/O	StringRange R/O
internal.output.03	TEXT	R/O	StringRange R/O
internal.output.04	TEXT	R/O	StringRange R/O
internal.output.05	TEXT	R/O	StringRange R/O
internal.output.06	TEXT	R/O	StringRange R/O
internal.output.07	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.output.08	TEXT	R/O	StringRange R/O
internal.output.09	TEXT	R/O	StringRange R/O
internal.output.10	TEXT	R/O	StringRange R/O
internal.output.11	TEXT	R/O	StringRange R/O
internal.output.12	TEXT	R/O	StringRange R/O
internal.output.13	TEXT	R/O	StringRange R/O
internal.output.14	TEXT	R/O	StringRange R/O
internal.output.15	TEXT	R/O	StringRange R/O
internal.output.16	TEXT	R/O	StringRange R/O
internal.output.17	TEXT	R/O	StringRange R/O
internal.output.18	TEXT	R/O	StringRange R/O
internal.output.19	TEXT	R/O	StringRange R/O
internal.output.20	TEXT	R/O	StringRange R/O
internal.output.21	TEXT	R/O	StringRange R/O
internal.output.22	TEXT	R/O	StringRange R/O
internal.output.23	TEXT	R/O	StringRange R/O
internal.output.24	TEXT	R/O	StringRange R/O
internal.output.25	TEXT	R/O	StringRange R/O
internal.output.26	TEXT	R/O	StringRange R/O
internal.output.27	TEXT	R/O	StringRange R/O
internal.output.28	TEXT	R/O	StringRange R/O
internal.output.29	TEXT	R/O	StringRange R/O
internal.output.30	TEXT	R/O	StringRange R/O
internal.output.31	TEXT	R/O	StringRange R/O
internal.output.32	TEXT	R/O	StringRange R/O
input.name.01	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
input.name.02	TEXT	R/O	StringRange R/O
input.name.03	TEXT	R/O	StringRange R/O
input.name.04	TEXT	R/O	StringRange R/O
input.name.05	TEXT	R/O	StringRange R/O
input.name.06	TEXT	R/O	StringRange R/O
input.name.07	TEXT	R/O	StringRange R/O
input.name.08	TEXT	R/O	StringRange R/O
input.name.09	TEXT	R/O	StringRange R/O
input.name.10	TEXT	R/O	StringRange R/O
input.name.11	TEXT	R/O	StringRange R/O
input.name.12	TEXT	R/O	StringRange R/O
input.name.13	TEXT	R/O	StringRange R/O
input.name.14	TEXT	R/O	StringRange R/O
input.name.15	TEXT	R/O	StringRange R/O
input.name.16	TEXT	R/O	StringRange R/O
input.name.17	TEXT	R/O	StringRange R/O
input.name.18	TEXT	R/O	StringRange R/O
input.name.19	TEXT	R/O	StringRange R/O
input.name.20	TEXT	R/O	StringRange R/O
input.name.21	TEXT	R/O	StringRange R/O
input.name.22	TEXT	R/O	StringRange R/O
input.name.23	TEXT	R/O	StringRange R/O
input.name.24	TEXT	R/O	StringRange R/O
input.name.25	TEXT	R/O	StringRange R/O
input.name.26	TEXT	R/O	StringRange R/O
input.name.27	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
input.name.28	TEXT	R/O	StringRange R/O
input.name.29	TEXT	R/O	StringRange R/O
input.name.30	TEXT	R/O	StringRange R/O
input.name.31	TEXT	R/O	StringRange R/O
input.name.32	TEXT	R/O	StringRange R/O
output.name.01	TEXT	R/O	StringRange R/O
output.name.02	TEXT	R/O	StringRange R/O
output.name.03	TEXT	R/O	StringRange R/O
output.name.04	TEXT	R/O	StringRange R/O
output.name.05	TEXT	R/O	StringRange R/O
output.name.06	TEXT	R/O	StringRange R/O
output.name.07	TEXT	R/O	StringRange R/O
output.name.08	TEXT	R/O	StringRange R/O
output.name.09	TEXT	R/O	StringRange R/O
output.name.10	TEXT	R/O	StringRange R/O
output.name.11	TEXT	R/O	StringRange R/O
output.name.12	TEXT	R/O	StringRange R/O
output.name.13	TEXT	R/O	StringRange R/O
output.name.14	TEXT	R/O	StringRange R/O
output.name.15	TEXT	R/O	StringRange R/O
output.name.16	TEXT	R/O	StringRange R/O
output.name.17	TEXT	R/O	StringRange R/O
output.name.18	TEXT	R/O	StringRange R/O
output.name.19	TEXT	R/O	StringRange R/O
output.name.20	TEXT	R/O	StringRange R/O
output.name.21	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.22	TEXT	R/O	StringRange R/O
output.name.23	TEXT	R/O	StringRange R/O
output.name.24	TEXT	R/O	StringRange R/O
output.name.25	TEXT	R/O	StringRange R/O
output.name.26	TEXT	R/O	StringRange R/O
output.name.27	TEXT	R/O	StringRange R/O
output.name.28	TEXT	R/O	StringRange R/O
output.name.29	TEXT	R/O	StringRange R/O
output.name.30	TEXT	R/O	StringRange R/O
output.name.31	TEXT	R/O	StringRange R/O
output.name.32	TEXT	R/O	StringRange R/O
reset	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Input 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Input 4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Input 5)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input 6)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Input 7)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 7)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Input 8)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Input 9)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Input 10)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Input 11)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Input 12)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Input 13)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Input 14)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Input 15)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Output 15)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Input 16)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Output 16)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Input 17)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Output 17)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Input 18)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Output 18)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Input 19)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Output 19)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Input 20)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Output 20)

name	*type*	*flags*	*range*
faults.41	ALARM	R/O	AlarmFlagRange R/O (Input 21)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Output 21)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Input 22)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Output 22)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Input 23)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Output 23)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Input 24)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Output 24)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Input 25)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Output 25)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Input 26)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Output 26)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Input 27)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Output 27)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Input 28)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Output 28)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Input 29)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Output 29)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Input 30)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Output 30)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Input 31)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Output 31)
faults.63	ALARM	R/O	AlarmFlagRange R/O (Input 32)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Output 32)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)

name	*type*	*flags*	*range*
faults.67	ALARM	R/O	AlarmFlagRange R/O (Power supply 3)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Power supply 4)
internal.changed	BOOLEAN	R/O	BooleanRange R/O (true false)

9.612 SatService-Power-Sensor

This page describes the device driver and the device window for the sat-nms Power Sensor.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the power reading and lets you set the power sensor's operational parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

parameter	*description*
address	This is the only parameter that must be set. Enter the power sensor's IP address in 'dotted quad' notation here. Example: " **192.168.2.81** ".

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.note	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.adcv	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.sensivity	TEXT	R/O	StringRange R/O
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
range	CHOICE		EnumRange (AUTO LOW-SENS HIGH-SENS)
frequency	INTEGER		IntegerRange (0 .. 19000)
offset	FLOAT		DoubleRange (-99.99 .. 99.99)
averaging	CHOICE		EnumRange (OFF FAST SLOW)
threshold	FLOAT		DoubleRange (-99.99 .. 99.99)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Level threshold)

9.613 SatService-SMU

This page describes the device driver and the device window for the SatService Signal Management Unit. It supports all available modules:

- sat-nms UMPS Universal Multipurpose Switch
- sat-nms TMPS Transfer Multipurpose Switch
- sat-nms LDCI Line Amplifier DC Inserter

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch position.
- 1 --- Slot 1 parameters.
- 2 --- Slot 2 parameters.

- 3 --- Slot 3 parameters.
- 4 --- Slot 4 parameters.
- 5 --- Slot 5 parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and 'private' as write community.

The read/write-community strings may be set through the web interface of the device. Factory defaults are 'public' for the read community, 'private' for the write community. Hence a typical address specification for this unit may be *10.0.0.1 public private* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
stateCycle	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
slot1.type	TEXT	R/O	StringRange R/O
slot1.description	TEXT	R/O	StringRange R/O
slot1.faultDelay	INTEGER		IntegerRange (0 .. 15)
slot1.cardFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.cardMissing	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.name	TEXT		StringRange
slot1.block1.position	CHOICE		EnumRange (A B)
slot1.block1.mode	CHOICE		EnumRange (MANUAL)
slot1.block1.dcPower	CHOICE		EnumRange (OFF ON)
slot1.block1.dcPowerA	CHOICE		EnumRange (OFF ON)
slot1.block1.dcPowerB	CHOICE		EnumRange (OFF ON)
slot1.block1.atten	INTEGER		IntegerRange (0 .. 31)
slot1.block1.thresh	INTEGER		IntegerRange (-99 .. -5)
slot1.block1.threshA	INTEGER		IntegerRange (-99 .. -5)
slot1.block1.threshB	INTEGER		IntegerRange (-99 .. -5)
slot1.block1.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block1.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot1.block1.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot1.block1.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block1.minCurr	INTEGER		IntegerRange (0 .. 250)
slot1.block1.minCurrA	INTEGER		IntegerRange (0 .. 250)

name	*type*	*flags*	*range*
slot1.block1.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot1.block1.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot1.block1.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot1.block1.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot1.block1.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block1.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.name	TEXT		StringRange
slot1.block2.position	CHOICE		EnumRange (A B)
slot1.block2.mode	CHOICE		EnumRange (MANUAL)
slot1.block2.dcPower	CHOICE		EnumRange (OFF ON)
slot1.block2.dcPowerA	CHOICE		EnumRange (OFF ON)
slot1.block2.dcPowerB	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot1.block2.atten	INTEGER		IntegerRange (0 .. 31)
slot1.block2.thresh	INTEGER		IntegerRange (-99 .. -5)
slot1.block2.threshA	INTEGER		IntegerRange (-99 .. -5)
slot1.block2.threshB	INTEGER		IntegerRange (-99 .. -5)
slot1.block2.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block2.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot1.block2.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot1.block2.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block2.minCurr	INTEGER		IntegerRange (0 .. 250)
slot1.block2.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot1.block2.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot1.block2.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot1.block2.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot1.block2.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot1.block2.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot1.block2.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block2.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.name	TEXT		StringRange
slot1.block3.position	CHOICE		EnumRange (A B)
slot1.block3.mode	CHOICE		EnumRange (MANUAL)
slot1.block3.dcPower	CHOICE		EnumRange (OFF ON)
slot1.block3.dcPowerA	CHOICE		EnumRange (OFF ON)
slot1.block3.dcPowerB	CHOICE		EnumRange (OFF ON)
slot1.block3.atten	INTEGER		IntegerRange (0 .. 31)
slot1.block3.thresh	INTEGER		IntegerRange (-99 .. -5)
slot1.block3.threshA	INTEGER		IntegerRange (-99 .. -5)
slot1.block3.threshB	INTEGER		IntegerRange (-99 .. -5)
slot1.block3.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block3.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot1.block3.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot1.block3.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block3.minCurr	INTEGER		IntegerRange (0 .. 250)
slot1.block3.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot1.block3.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot1.block3.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot1.block3.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot1.block3.maxCurrB	INTEGER		IntegerRange (50 .. 450)

name	*type*	*flags*	*range*
slot1.block3.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block3.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block3.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.name	TEXT		StringRange
slot1.block4.position	CHOICE		EnumRange (A B)
slot1.block4.mode	CHOICE		EnumRange (MANUAL)
slot1.block4.dcPower	CHOICE		EnumRange (OFF ON)
slot1.block4.dcPowerA	CHOICE		EnumRange (OFF ON)
slot1.block4.dcPowerB	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot1.block4.atten	INTEGER		IntegerRange (0 .. 31)
slot1.block4.thresh	INTEGER		IntegerRange (-99 .. -5)
slot1.block4.threshA	INTEGER		IntegerRange (-99 .. -5)
slot1.block4.threshB	INTEGER		IntegerRange (-99 .. -5)
slot1.block4.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block4.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot1.block4.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot1.block4.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot1.block4.minCurr	INTEGER		IntegerRange (0 .. 250)
slot1.block4.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot1.block4.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot1.block4.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot1.block4.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot1.block4.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot1.block4.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot1.block4.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot1.block4.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot1.block4.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.type	TEXT	R/O	StringRange R/O
slot2.description	TEXT	R/O	StringRange R/O
slot2.faultDelay	INTEGER		IntegerRange (0 .. 15)
slot2.cardFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.cardMissing	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.name	TEXT		StringRange
slot2.block1.position	CHOICE		EnumRange (A B)
slot2.block1.mode	CHOICE		EnumRange (MANUAL)
slot2.block1.dcPower	CHOICE		EnumRange (OFF ON)
slot2.block1.dcPowerA	CHOICE		EnumRange (OFF ON)
slot2.block1.dcPowerB	CHOICE		EnumRange (OFF ON)
slot2.block1.atten	INTEGER		IntegerRange (0 .. 31)
slot2.block1.thresh	INTEGER		IntegerRange (-99 .. -5)
slot2.block1.threshA	INTEGER		IntegerRange (-99 .. -5)
slot2.block1.threshB	INTEGER		IntegerRange (-99 .. -5)
slot2.block1.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block1.iOffsA	INTEGER		IntegerRange (-10 .. 10)

name	*type*	*flags*	*range*
slot2.block1.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot2.block1.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block1.minCurr	INTEGER		IntegerRange (0 .. 250)
slot2.block1.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot2.block1.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot2.block1.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot2.block1.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot2.block1.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot2.block1.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block1.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot2.block2.name	TEXT		StringRange
slot2.block2.position	CHOICE		EnumRange (A B)
slot2.block2.mode	CHOICE		EnumRange (MANUAL)
slot2.block2.dcPower	CHOICE		EnumRange (OFF ON)
slot2.block2.dcPowerA	CHOICE		EnumRange (OFF ON)
slot2.block2.dcPowerB	CHOICE		EnumRange (OFF ON)
slot2.block2.atten	INTEGER		IntegerRange (0 .. 31)
slot2.block2.thresh	INTEGER		IntegerRange (-99 .. -5)
slot2.block2.threshA	INTEGER		IntegerRange (-99 .. -5)
slot2.block2.threshB	INTEGER		IntegerRange (-99 .. -5)
slot2.block2.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block2.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot2.block2.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot2.block2.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block2.minCurr	INTEGER		IntegerRange (0 .. 250)
slot2.block2.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot2.block2.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot2.block2.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot2.block2.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot2.block2.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot2.block2.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot2.block2.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block2.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.name	TEXT		StringRange
slot2.block3.position	CHOICE		EnumRange (A B)
slot2.block3.mode	CHOICE		EnumRange (MANUAL)
slot2.block3.dcPower	CHOICE		EnumRange (OFF ON)
slot2.block3.dcPowerA	CHOICE		EnumRange (OFF ON)
slot2.block3.dcPowerB	CHOICE		EnumRange (OFF ON)
slot2.block3.atten	INTEGER		IntegerRange (0 .. 31)
slot2.block3.thresh	INTEGER		IntegerRange (-99 .. -5)
slot2.block3.threshA	INTEGER		IntegerRange (-99 .. -5)
slot2.block3.threshB	INTEGER		IntegerRange (-99 .. -5)
slot2.block3.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block3.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot2.block3.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot2.block3.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block3.minCurr	INTEGER		IntegerRange (0 .. 250)
slot2.block3.minCurrA	INTEGER		IntegerRange (0 .. 250)

name	*type*	*flags*	*range*
slot2.block3.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot2.block3.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot2.block3.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot2.block3.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot2.block3.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block3.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block3.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot2.block4.name	TEXT		StringRange
slot2.block4.position	CHOICE		EnumRange (A B)
slot2.block4.mode	CHOICE		EnumRange (MANUAL)
slot2.block4.dcPower	CHOICE		EnumRange (OFF ON)
slot2.block4.dcPowerA	CHOICE		EnumRange (OFF ON)
slot2.block4.dcPowerB	CHOICE		EnumRange (OFF ON)
slot2.block4.atten	INTEGER		IntegerRange (0 .. 31)
slot2.block4.thresh	INTEGER		IntegerRange (-99 .. -5)
slot2.block4.threshA	INTEGER		IntegerRange (-99 .. -5)
slot2.block4.threshB	INTEGER		IntegerRange (-99 .. -5)
slot2.block4.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block4.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot2.block4.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot2.block4.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot2.block4.minCurr	INTEGER		IntegerRange (0 .. 250)
slot2.block4.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot2.block4.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot2.block4.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot2.block4.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot2.block4.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot2.block4.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot2.block4.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot2.block4.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot2.block4.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.type	TEXT	R/O	StringRange R/O
slot3.description	TEXT	R/O	StringRange R/O
slot3.faultDelay	INTEGER		IntegerRange (0 .. 15)
slot3.cardFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.cardMissing	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.name	TEXT		StringRange
slot3.block1.position	CHOICE		EnumRange (A B)
slot3.block1.mode	CHOICE		EnumRange (MANUAL)
slot3.block1.dcPower	CHOICE		EnumRange (OFF ON)
slot3.block1.dcPowerA	CHOICE		EnumRange (OFF ON)
slot3.block1.dcPowerB	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot3.block1.atten	INTEGER		IntegerRange (0 .. 31)
slot3.block1.thresh	INTEGER		IntegerRange (-99 .. -5)
slot3.block1.threshA	INTEGER		IntegerRange (-99 .. -5)
slot3.block1.threshB	INTEGER		IntegerRange (-99 .. -5)
slot3.block1.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block1.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot3.block1.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot3.block1.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block1.minCurr	INTEGER		IntegerRange (0 .. 250)
slot3.block1.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot3.block1.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot3.block1.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot3.block1.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot3.block1.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot3.block1.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block1.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block1.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot3.block1.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block1.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block1.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block1.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.name	TEXT		StringRange
slot3.block2.position	CHOICE		EnumRange (A B)
slot3.block2.mode	CHOICE		EnumRange (MANUAL)
slot3.block2.dcPower	CHOICE		EnumRange (OFF ON)
slot3.block2.dcPowerA	CHOICE		EnumRange (OFF ON)
slot3.block2.dcPowerB	CHOICE		EnumRange (OFF ON)
slot3.block2.atten	INTEGER		IntegerRange (0 .. 31)
slot3.block2.thresh	INTEGER		IntegerRange (-99 .. -5)
slot3.block2.threshA	INTEGER		IntegerRange (-99 .. -5)
slot3.block2.threshB	INTEGER		IntegerRange (-99 .. -5)
slot3.block2.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block2.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot3.block2.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot3.block2.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block2.minCurr	INTEGER		IntegerRange (0 .. 250)
slot3.block2.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot3.block2.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot3.block2.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot3.block2.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot3.block2.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot3.block2.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot3.block2.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block2.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.name	TEXT		StringRange
slot3.block3.position	CHOICE		EnumRange (A B)
slot3.block3.mode	CHOICE		EnumRange (MANUAL)
slot3.block3.dcPower	CHOICE		EnumRange (OFF ON)
slot3.block3.dcPowerA	CHOICE		EnumRange (OFF ON)
slot3.block3.dcPowerB	CHOICE		EnumRange (OFF ON)
slot3.block3.atten	INTEGER		IntegerRange (0 .. 31)
slot3.block3.thresh	INTEGER		IntegerRange (-99 .. -5)
slot3.block3.threshA	INTEGER		IntegerRange (-99 .. -5)
slot3.block3.threshB	INTEGER		IntegerRange (-99 .. -5)

name	*type*	*flags*	*range*
slot3.block3.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block3.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot3.block3.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot3.block3.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block3.minCurr	INTEGER		IntegerRange (0 .. 250)
slot3.block3.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot3.block3.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot3.block3.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot3.block3.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot3.block3.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot3.block3.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block3.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot3.block3.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block3.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.name	TEXT		StringRange
slot3.block4.position	CHOICE		EnumRange (A B)
slot3.block4.mode	CHOICE		EnumRange (MANUAL)
slot3.block4.dcPower	CHOICE		EnumRange (OFF ON)
slot3.block4.dcPowerA	CHOICE		EnumRange (OFF ON)
slot3.block4.dcPowerB	CHOICE		EnumRange (OFF ON)
slot3.block4.atten	INTEGER		IntegerRange (0 .. 31)
slot3.block4.thresh	INTEGER		IntegerRange (-99 .. -5)
slot3.block4.threshA	INTEGER		IntegerRange (-99 .. -5)
slot3.block4.threshB	INTEGER		IntegerRange (-99 .. -5)
slot3.block4.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block4.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot3.block4.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot3.block4.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot3.block4.minCurr	INTEGER		IntegerRange (0 .. 250)
slot3.block4.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot3.block4.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot3.block4.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot3.block4.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot3.block4.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot3.block4.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot3.block4.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot3.block4.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot3.block4.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.type	TEXT	R/O	StringRange R/O
slot4.description	TEXT	R/O	StringRange R/O
slot4.faultDelay	INTEGER		IntegerRange (0 .. 16)
slot4.cardFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.cardMissing	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.name	TEXT		StringRange
slot4.block1.position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
slot4.block1.mode	CHOICE		EnumRange (MANUAL)
slot4.block1.dcPower	CHOICE		EnumRange (OFF ON)
slot4.block1.dcPowerA	CHOICE		EnumRange (OFF ON)
slot4.block1.dcPowerB	CHOICE		EnumRange (OFF ON)
slot4.block1.atten	INTEGER		IntegerRange (0 .. 31)
slot4.block1.thresh	INTEGER		IntegerRange (-99 .. -5)
slot4.block1.threshA	INTEGER		IntegerRange (-99 .. -5)
slot4.block1.threshB	INTEGER		IntegerRange (-99 .. -5)
slot4.block1.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block1.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot4.block1.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot4.block1.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block1.minCurr	INTEGER		IntegerRange (0 .. 250)
slot4.block1.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot4.block1.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot4.block1.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot4.block1.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot4.block1.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot4.block1.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot4.block1.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block1.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.name	TEXT		StringRange
slot4.block2.position	CHOICE		EnumRange (A B)
slot4.block2.mode	CHOICE		EnumRange (MANUAL)
slot4.block2.dcPower	CHOICE		EnumRange (OFF ON)
slot4.block2.dcPowerA	CHOICE		EnumRange (OFF ON)
slot4.block2.dcPowerB	CHOICE		EnumRange (OFF ON)
slot4.block2.atten	INTEGER		IntegerRange (0 .. 31)
slot4.block2.thresh	INTEGER		IntegerRange (-99 .. -5)
slot4.block2.threshA	INTEGER		IntegerRange (-99 .. -5)
slot4.block2.threshB	INTEGER		IntegerRange (-99 .. -5)
slot4.block2.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block2.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot4.block2.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot4.block2.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block2.minCurr	INTEGER		IntegerRange (0 .. 250)
slot4.block2.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot4.block2.minCurrB	INTEGER		IntegerRange (0 .. 250)

name	*type*	*flags*	*range*
slot4.block2.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot4.block2.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot4.block2.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot4.block2.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block2.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.name	TEXT		StringRange
slot4.block3.position	CHOICE		EnumRange (A B)
slot4.block3.mode	CHOICE		EnumRange (MANUAL)
slot4.block3.dcPower	CHOICE		EnumRange (OFF ON)
slot4.block3.dcPowerA	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot4.block3.dcPowerB	CHOICE		EnumRange (OFF ON)
slot4.block3.atten	INTEGER		IntegerRange (0 .. 31)
slot4.block3.thresh	INTEGER		IntegerRange (-99 .. -5)
slot4.block3.threshA	INTEGER		IntegerRange (-99 .. -5)
slot4.block3.threshB	INTEGER		IntegerRange (-99 .. -5)
slot4.block3.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block3.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot4.block3.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot4.block3.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block3.minCurr	INTEGER		IntegerRange (0 .. 250)
slot4.block3.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot4.block3.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot4.block3.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot4.block3.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot4.block3.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot4.block3.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot4.block3.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block3.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block3.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.name	TEXT		StringRange
slot4.block4.position	CHOICE		EnumRange (A B)
slot4.block4.mode	CHOICE		EnumRange (MANUAL)
slot4.block4.dcPower	CHOICE		EnumRange (OFF ON)
slot4.block4.dcPowerA	CHOICE		EnumRange (OFF ON)
slot4.block4.dcPowerB	CHOICE		EnumRange (OFF ON)
slot4.block4.atten	INTEGER		IntegerRange (0 .. 31)
slot4.block4.thresh	INTEGER		IntegerRange (-99 .. -5)
slot4.block4.threshA	INTEGER		IntegerRange (-99 .. -5)
slot4.block4.threshB	INTEGER		IntegerRange (-99 .. -5)
slot4.block4.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block4.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot4.block4.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot4.block4.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot4.block4.minCurr	INTEGER		IntegerRange (0 .. 250)
slot4.block4.minCurrA	INTEGER		IntegerRange (0 .. 250)

name	*type*	*flags*	*range*
slot4.block4.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot4.block4.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot4.block4.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot4.block4.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot4.block4.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot4.block4.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot4.block4.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot5.description	TEXT	R/O	StringRange R/O
slot5.faultDelay	INTEGER		IntegerRange (0 .. 15)
slot5.cardFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.cardMissing	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.name	TEXT		StringRange
slot5.block1.position	CHOICE		EnumRange (A B)
slot5.block1.mode	CHOICE		EnumRange (MANUAL)
slot5.block1.dcPower	CHOICE		EnumRange (OFF ON)
slot5.block1.dcPowerA	CHOICE		EnumRange (OFF ON)
slot5.block1.dcPowerB	CHOICE		EnumRange (OFF ON)
slot5.block1.atten	INTEGER		IntegerRange (0 .. 31)
slot5.block1.thresh	INTEGER		IntegerRange (-99 .. -5)
slot5.block1.threshA	INTEGER		IntegerRange (-99 .. -5)
slot5.block1.threshB	INTEGER		IntegerRange (-99 .. -5)
slot5.block1.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block1.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot5.block1.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot5.block1.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block1.minCurr	INTEGER		IntegerRange (0 .. 250)
slot5.block1.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot5.block1.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot5.block1.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot5.block1.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot5.block1.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot5.block1.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot5.block1.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block1.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block1.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.name	TEXT		StringRange
slot5.block2.position	CHOICE		EnumRange (A B)
slot5.block2.mode	CHOICE		EnumRange (MANUAL)
slot5.block2.dcPower	CHOICE		EnumRange (OFF ON)
slot5.block2.dcPowerA	CHOICE		EnumRange (OFF ON)
slot5.block2.dcPowerB	CHOICE		EnumRange (OFF ON)
slot5.block2.atten	INTEGER		IntegerRange (0 .. 31)
slot5.block2.thresh	INTEGER		IntegerRange (-99 .. -5)
slot5.block2.threshA	INTEGER		IntegerRange (-99 .. -5)
slot5.block2.threshB	INTEGER		IntegerRange (-99 .. -5)
slot5.block2.iOffs	INTEGER		IntegerRange (-10 .. 10)

name	*type*	*flags*	*range*
slot5.block2.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot5.block2.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot5.block2.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block2.minCurr	INTEGER		IntegerRange (0 .. 250)
slot5.block2.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot5.block2.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot5.block2.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot5.block2.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot5.block2.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot5.block2.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block2.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block2.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.name	TEXT		StringRange

name	*type*	*flags*	*range*
slot5.block3.position	CHOICE		EnumRange (A B)
slot5.block3.mode	CHOICE		EnumRange (MANUAL)
slot5.block3.dcPower	CHOICE		EnumRange (OFF ON)
slot5.block3.dcPowerA	CHOICE		EnumRange (OFF ON)
slot5.block3.dcPowerB	CHOICE		EnumRange (OFF ON)
slot5.block3.atten	INTEGER		IntegerRange (0 .. 31)
slot5.block3.thresh	INTEGER		IntegerRange (-99 .. -5)
slot5.block3.threshA	INTEGER		IntegerRange (-99 .. -5)
slot5.block3.threshB	INTEGER		IntegerRange (-99 .. -5)
slot5.block3.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block3.iOffsA	INTEGER		IntegerRange (-10 .. 10)
slot5.block3.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot5.block3.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block3.minCurr	INTEGER		IntegerRange (0 .. 250)
slot5.block3.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot5.block3.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot5.block3.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot5.block3.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot5.block3.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot5.block3.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot5.block3.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block3.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block3.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.name	TEXT		StringRange
slot5.block4.position	CHOICE		EnumRange (A B)
slot5.block4.mode	CHOICE		EnumRange (MANUAL)
slot5.block4.dcPower	CHOICE		EnumRange (OFF ON)
slot5.block4.dcPowerA	CHOICE		EnumRange (OFF ON)
slot5.block4.dcPowerB	CHOICE		EnumRange (OFF ON)
slot5.block4.atten	INTEGER		IntegerRange (0 .. 31)
slot5.block4.thresh	INTEGER		IntegerRange (-99 .. -5)
slot5.block4.threshA	INTEGER		IntegerRange (-99 .. -5)
slot5.block4.threshB	INTEGER		IntegerRange (-99 .. -5)
slot5.block4.iOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block4.iOffsA	INTEGER		IntegerRange (-10 .. 10)

name	*type*	*flags*	*range*
slot5.block4.iOffsB	INTEGER		IntegerRange (-10 .. 10)
slot5.block4.oOffs	INTEGER		IntegerRange (-10 .. 10)
slot5.block4.minCurr	INTEGER		IntegerRange (0 .. 250)
slot5.block4.minCurrA	INTEGER		IntegerRange (0 .. 250)
slot5.block4.minCurrB	INTEGER		IntegerRange (0 .. 250)
slot5.block4.maxCurr	INTEGER		IntegerRange (50 .. 450)
slot5.block4.maxCurrA	INTEGER		IntegerRange (50 .. 450)
slot5.block4.maxCurrB	INTEGER		IntegerRange (50 .. 450)
slot5.block4.loCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.loCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.loCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.hiCurr	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.hiCurrA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.hiCurrB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.lowRf	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.lowRfA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.lowRfB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.swFault	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.inFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.inFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.slFaultA	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.slFaultB	BOOLEAN	R/O	BooleanRange R/O (true false)
slot5.block4.current	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.currentA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.currentB	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
slot5.block4.iLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.iLevelA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.iLevelB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot5.block4.oLevel	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Card 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Block 1.1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Block 1.2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Block 1.3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Block 1.4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Card 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Block 2.1)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block 2.2)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Block 2.3)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Block 2.4)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Card 3)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Block 3.1)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Block 3.2)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Block 3.3)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Block 3.4)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Card 4)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Block 4.1)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Block 4.2)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Block 4.3)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Block 4.4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Card 5)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Block 5.1)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Block 5.2)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Block 5.3)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Block 5.4)
faults.26	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.27	ALARM	R/O	AlarmFlagRange R/O (PSU 2)

9.614 SatService-Temp-232

This page describes the device driver and the device window for the*SatService-Temp-232* device. This device reads the temperature value from the Temp-RS232 temperature sensor.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual temperature value.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the alarm monitoring input at the PLC.

parameter	*description*
address	Leave this field empty, the Temp-RS232 sensor does not require a device address.
Serial	Remember to set the line parameters for the serial interface to*9600,N,8,1* , no handshaking.

Remarks

- The sensor is powered through the handshake lines of the serial interface. If you need to extend the interface cable, either use an off the shelf 9-pole extension cable or at least connect the GND, TxD, RxD RTS and DTR lines.
- The sensor is not polled by the software, it autonomously sends a temperature value every second. For best results, set the IDLE time for this interface to 200ms in the VLC

setup dialog.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hiThreshold	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
loThreshold	FLOAT	SAVE	DoubleRange (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature LO threshold)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Temperature HI threshold)

9.615 Satellink-1to2-Redundant-LNA

This page describes the device driver and the device window for the Satellink-1to2-Redundant-LNA controller.

The sat-nms M&C communicates with this device through a built in Lantronix XPort IP to RS232 converter. The IP address of this device has to be set as the interface name in the sat-nms device setup, the factory default for the port number is 10001.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the redundancy switching parameters
- --- Contains the LNA/LNB meter readings (not applicable with this device).

Configuration parameters

There are no special configuration parameters for this device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
internal.p	TEXT	R/O	StringRange R/O
protection	CHOICE		EnumRange (MANUAL AUTO)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B ?)
position.2	CHOICE		EnumRange (A B ?)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Remote control)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (LNA/LNB3)

9.616 SatelliteSystems-3430-BTR-L

This page describes the device driver and the device window for the Satellite Systems (SSC) L-Band beacon tracking receiver of series 3430 BTR-L . It uses the Terminal-CR-CR protocol.

- Default TCP communication port is 8080

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	Device address for serial communication
Serial	Remember to set the line parameters for the serial interface (only if connected to serial port of beacon receiver and not to Ethernet)
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
frequency	FLOAT		DoubleRange (930.00 .. 2300.00)
reference	FLOAT		DoubleRange (-100.0 .. -40.0)
internal.v	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Lock alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.617 Scopus-E1X00-SNMP

This page describes the device driver and the device window for the Scopus E1X00 DVB Encoder family, controlled via SNMP. Beside the basic E1000 encoder, this supports the E1720 encoder with integrated modulator.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- MPEG output parameters.
- --- BISS encryption parameters
- --- Video parameters.
- --- Audio parameters.
- --- Auxiliary data parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The only configuration parameter required for the E1000 encoder is the device address. Please note, that the E1000 encoder requires the SNMP community string to be set in special way. The encoder expects the community to be set to *E1000-<ip address>* . This community string must be appended to the IP address separated by a space character. Example:

192.168.1.20 E1000-192.168.1.20

The above is the valid address / community string setting for an encoder with the IP address 192.168.1.20.

Remarks

The E1720 encoder requires the output level setting to be 'latched in' with an encoder restart command like all the other parameters, too. Hence stepping the level up/down works very slow and may cause drop outs in the encoded video. The E1720 output level is not suitable for the use in a power control loop for this reason.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.bucLoFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
internal.changed	TEXT	R/O	StringRange R/O
internal.tmp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.model	TEXT	R/O	StringRange R/O
internal.family	TEXT	R/O	StringRange R/O
internal.lbfreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.type	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
info.model	TEXT	R/O	StringRange R/O
info.audioChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.srevision	TEXT	R/O	StringRange R/O
info.hrevision	TEXT	R/O	StringRange R/O
info.permcode	TEXT	R/O	StringRange R/O
info.videoifc	TEXT	R/O	StringRange R/O
info.busy	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
mod.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
mod.symbolrate	FLOAT		DoubleRange (0.00000 .. 0.00000)
mod.fec	CHOICE		EnumRange (1/2 QPSK 2/3 QPSK 3/4 QPSK 5/6 QPSK 7/8 QPSK 2/3 8PSK 3/4 16QAM)
mod.fec2	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8 2/3 3/4)
mod.modulation2	CHOICE		EnumRange (QPSK QPSK QPSK QPSK QPSK 8PSK 16QAM)
mod.spectrum	CHOICE		EnumRange (NORM INVERT)
mod.modulation	CHOICE		EnumRange (OFF ON)
mod.rfon	CHOICE		EnumRange (OFF ON)
mod.power	FLOAT		DoubleRange (0 .. 0)
mod.ucmPower	FLOAT		DoubleRange (0 .. 0)
mod.rollOff	CHOICE		EnumRange (35 % 25 % 20 % 15 %)
output.datarate	INTEGER		IntegerRange (1000000 .. 52000000)
output.vtest	CHOICE		EnumRange (OFF ON)
output.atest	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
output.clksource	CHOICE		EnumRange (INTERN EXTERN)
output.cascade	CHOICE		EnumRange (OFF ON)
output.statmux	CHOICE		EnumRange (OFF ON)
output.pmtpid	HEX		HexRange (20 .. 1FFF)
output.pcrpid	HEX		HexRange (0 .. 1FFF)
output.tid	HEX		HexRange (0 .. FFFF)
output.onid	HEX		HexRange (0 .. FFFF)
output.camode	CHOICE		EnumRange (CLEAR ENCODER CAS-5000 EBU BISS)
output.profile	CHOICE		EnumRange (4:2:0 4:2:0 LD 4:2:2 4:2:2 LD)
output.dvbtables	CHOICE		EnumRange (ELIMINATED NORMAL PLAY CONTENT)
output.packetsize	CHOICE		EnumRange (188 204)
output.program	HEX		HexRange (1 .. 1FFFF)
output.service	TEXT		StringRange
biss.mode	CHOICE		EnumRange (CLEAR BISS-1 BISS-E-INJECTED BISS-E-BURIED)
biss.injected.lo	HEX		HexRange (0 .. 0)
biss.injected.hi	HEX		HexRange (0 .. 0)
biss.clearsw.lo	HEX		HexRange (0 .. 0)
biss.clearsw.hi	HEX		HexRange (0 .. 0)
biss.cryptedsw.lo	HEX		HexRange (0 .. 0)
biss.cryptedsw.hi	HEX		HexRange (0 .. 0)
video.source	CHOICE		EnumRange (COMPOSITE S-VHS COMPONENT SDI)
video.format	CHOICE		EnumRange (NTSC PAL)
video.datarate	INTEGER		IntegerRange (1000000 .. 50000000)

name	*type*	*flags*	*range*
video.resolution	CHOICE		EnumRange (720 FULL 704 FULL 640 FULL 544 FULL 368 FULL 352 HALF 352 FULL 480 FULL AUTO)
video.pid	HEX		HexRange (20 .. 1FFF)
video.aspect	CHOICE		EnumRange (4:3 16:9 4:3 AUTO 16:9 AUTO)
video.dnr	CHOICE		EnumRange (OFF LOW MED HIGH)
video.luminance	INTEGER		IntegerRange (0 .. 255)
video.chrominance	INTEGER		IntegerRange (0 .. 255)
video.offset	INTEGER		IntegerRange (0 .. 255)
video.yc-delay	INTEGER		IntegerRange (0 .. 12)
video.phase	INTEGER		IntegerRange (0 .. 20)
video.chrphase	INTEGER		IntegerRange (0 .. 255)
video.admode	CHOICE		EnumRange (TV VCR DVD)
video.defaults	TEXT	NOPRESET	StringRange
audio.sampling	CHOICE		EnumRange (32 kHz 44.1 kHz 48 kHz)
audio.avsync	INTEGER		IntegerRange (-3000 .. 3000)
audio.1.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.1.source	CHOICE		EnumRange (ANALOG AES-EBU EMBEDDED AUTO DOLBY LINEAR)
audio.1.mode	CHOICE		EnumRange (STEREO JOINT DUAL MONO)
audio.1.pid	HEX		HexRange (20 .. 1FFF)
audio.1.lang	TEXT		StringRange
audio.1.volume	INTEGER		IntegerRange (-12 .. 12)
audio.2.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)

name	*type*	*flags*	*range*
audio.2.source	CHOICE		EnumRange (ANALOG AES-EBU EMBEDDED AUTO DOLBY LINEAR)
audio.2.mode	CHOICE		EnumRange (STEREO JOINT DUAL MONO)
audio.2.pid	HEX		HexRange (20 .. 1FFF)
audio.2.lang	TEXT		StringRange
audio.2.volume	INTEGER		IntegerRange (-12 .. 12)
audio.3.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.3.source	CHOICE		EnumRange (ANALOG AES-EBU EMBEDDED AUTO DOLBY LINEAR)
audio.3.mode	CHOICE		EnumRange (STEREO JOINT DUAL MONO)
audio.3.pid	HEX		HexRange (20 .. 1FFF)
audio.3.lang	TEXT		StringRange
audio.3.volume	INTEGER		IntegerRange (-12 .. 12)
audio.4.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.4.source	CHOICE		EnumRange (ANALOG AES-EBU EMBEDDED AUTO DOLBY LINEAR)
audio.4.mode	CHOICE		EnumRange (STEREO JOINT DUAL MONO)
audio.4.pid	HEX		HexRange (20 .. 1FFF)
audio.4.lang	TEXT		StringRange
audio.4.volume	INTEGER		IntegerRange (-12 .. 12)
auxdata.datarate	CHOICE		EnumRange (OFF 2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 57600 bps 115200 bps)
auxdata.pid	HEX		HexRange (20 .. 1FFF)

name	*type*	*flags*	*range*
hsdata.datarate	INTEGER		IntegerRange (0 .. 20000000)
hsdata.pid	HEX		HexRange (20 .. 1FFF)
hsdata.clock	CHOICE		EnumRange (64 kHz 512 kHz 1024 kHz 2048 kHz 3072 kHz 4096 kHz 6144 kHz 8192 kHz)
vbi.fields	CHOICE		EnumRange (ALL OFF TTX VPS VPS+TTX WSS WSS+TTX WSS+VPS WSS+VPS+TTX SMC SMC+TTX SMC+VPS SMC+VPS+TTX SMC+WSS SMC+WSS+TTX SMC+WSS+VPS SMC+WSS+VPS+TTX)
teletext.pid	HEX		HexRange (20 .. 1FFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Video codec)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Audio 1 codec)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 2 codec)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 3 codec)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 4 codec)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Multiplexer)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Modulator)

9.618 Scopus-IRD26XX

This page describes the device driver and the device window for Scopus 26XX IRDs. Actually the driver only supports the 'QPSK' front end module of the IRD, which is used for the satellite receive application.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters.
- --- Contains the parameters concerning the audio and video output (Analog video

format selection test frames).
- --- Contains the parameters concerning decoder input signal selection and the SDI digital video output.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Scopus IRD.

parameter	***description***
address	You must set the device address of the IRD here to make the program talk to the device. The device address is a decimal number in the range 129 to 255. The factory preset address at the IRDs is "129" but this may be set to other values at the IRD front panel.
lbandInputs	Select the number of L-band inputs the receiver provides (depending on the model 1 or 2 inputs)
22kHz	select if the IRD shall apply a 22kHz tone to the LNB supply voltage for input frequencies above 'bandEdge'.
lnbPower	Select if the IRD shall apply LNB supply voltage. You may select 14V or 18V to make a LNB which supports this change the polarization
loFreqHigh	Enter the LO frequency for the upper frequency band here. This does not set the LO frequency used by the IRD (see below).
loFreqLow	Enter the LO frequency for the lower frequency band here. This does not set the LO frequency used by the IRD (see below).
bandEdge	Enter frequency here, where the receiver shall switch to the upper frequency band.
audio.1	Select the hardware installed for the first audio channel here.
audio.2	Select the hardware installed for the second audio channel here.
audio.3	Select the hardware installed for the third audio channel here.
ebnoThreshold	The Eb/No threshold value. The driver raises an alarm if the Eb/No ration falls below this value.

parameter	***description***
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the IRD. The recommended setting is *19200,N,8,1* .

Preparing the IRD for remote control

Some settings at Scopus IRD's font panel must be made before the IRD can be connected to the M&C system. These settings are:

- Master-Slave protocol type
- Unit address 129 (recommended value)
- Remote interface RS232 (unless you are using the RS485 hardware).
- Remote interface baudrate 19200 (recommended value)
- C/Ku Band frequency display (Menue "CONFIG RECEIVER", see below).

L-band vs. RF frequency display

The Scopus IRD is capable to set/view the receive frequency either in terms of L-band or true RF receive frequency. It however does not support the explicit programming of the LNB LO frequencies. To make the IRD accept and show true RF frequencies, you have to perform the following steps:

1. At the IRD front panel, go to the CONFIG RECEIVER menu and select the "satellite frequency" option which matches the LO frequencies your antenna uses.
2. At the sat-nms software, open the maintenance / setup dialog for the IRD and set the *loFreqHigh* , *loFreqLow* and *bandEdge* parameter to the proper frequencies. These values do not influence the receiver's display, but they are used by the driver to control the 22kHz tone while the *22Khz* parameter is set to *AUTO* .

If you want to display the receive frequency setting as a L-band value, set this at the receiver's front panel and the *loFreqHigh* , *loFreqLow* and *bandEdge* parameters to zero.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.frontend	TEXT	R/O	StringRange R/O
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (1 2)
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF AUTO)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 14V-Vertical 18V-Horizontal)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.audio.1	CHOICE	SAVE SETUP	EnumRange (OFF ANALOG DIGITAL)
config.audio.2	CHOICE	SAVE SETUP	EnumRange (OFF ANALOG DIGITAL)
config.ebnoThreshold	FLOAT	SAVE SETUP	DoubleRange (1.0 .. 25.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.numServices	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ack	TEXT	R/O	StringRange R/O
internal.flags	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mustBoot	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.bootTimer	TEXT	R/O	StringRange R/O
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.loFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fec	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange DIS (A B)
polarization	CHOICE		EnumRange (X Y)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
symbolRate	FLOAT		DoubleRange (1.00000 .. 44.00000)
modulation	CHOICE		EnumRange (QPSK)
fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 AUTO)
programNo	INTEGER		IntegerRange (0 .. 0)
programList	TEXT	R/O	StringRange R/O
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.format	CHOICE		EnumRange (NTSC PAL BG PAL M PAL N SECAM PAL D)
video.aspect	CHOICE		EnumRange (4:3 16:9 AUTO OFF)
video.test	CHOICE		EnumRange (OFF NTSC PAL BG PAL M PAL N SECAM PAL D)
audio.1.program	INTEGER		IntegerRange DIS (0 .. 99)
audio.1.routing	CHOICE		EnumRange DIS (OFF ON)
audio.1.output	CHOICE		EnumRange DIS (MONO STEREO)
audio.1.level	CHOICE		EnumRange DIS ()

name	*type*	*flags*	*range*
audio.2.program	INTEGER		IntegerRange DIS (0 .. 99)
audio.2.routing	CHOICE		EnumRange DIS (OFF ON)
audio.2.level	CHOICE		EnumRange DIS ()
audio.format	CHOICE		EnumRange (MPEG AC3 LINEAR)
audio.ac3rate	CHOICE		EnumRange (32 kHz 44.1 kHz 48 kHz)
unitType	CHOICE		EnumRange (IRD DECODER)
decoderInput	CHOICE		EnumRange (SPI ASI RS422 INTERNAL)
digitalOutput	CHOICE		EnumRange (OFF ON)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mpeg	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ebno	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.signal	INTEGER	R/O	IntegerRange R/O (0 .. 100)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.aspect	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (DSNG BISS-1 BISS-E-CLR BISS-E-INJ BISS-E-BUR)
ca.bisssession	TEXT		StringRange
ca.bissinjected	TEXT		StringRange
ca.dsngoddkey	TEXT		StringRange
ca.dsngevenkey	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Frontend not supported)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Signal lock)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (MPEG decode)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Eb/No)

9.619 Scopus-IRD28XX

This page describes the device driver and the device window for Scopus 28XX IRDs. Actually the driver supports the 'QPSK' and the "DSNG" front end modules of the IRD, which is used for the satellite receive applications.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters.
- --- Contains the parameters concerning the audio and video output (Analog video format selection test frames).
- --- Contains the parameters concerning decoder input signal selection and the SDI digital video output.
- --- Contains the meter reading.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Scopus IRD.

parameter	*description*
address	You must set the device address of the IRD here to make the program talk to the device. The device address is a decimal number in the range 129 to 255. The factory preset address at the IRDs is "129" but this may be set to other values at the IRD front panel.
lbandInputs	Select the number of L-band inputs the receiver provides (depending on the model 1 or 2 inputs)
22kHz	select if the IRD shall apply a 22kHz tone to the LNB supply voltage for input frequencies above 'bandEdge'.

parameter	***description***
lnbPower	Select if the IRD shall apply LNB supply voltage. You may select 14V or 18V to make a LNB which supports this change the polarization
loFreqHigh	Enter the LO frequency for the upper frequency band here. This does not set the LO frequency used by the IRD (see below).
loFreqLow	Enter the LO frequency for the lower frequency band here. This does not set the LO frequency used by the IRD (see below).
bandEdge	Enter frequency here, where the receiver shall switch to the upper frequency band.
audio.1	Select the hardware installed for the first audio channel here.
audio.2	Select the hardware installed for the second audio channel here.
audio.3	Select the hardware installed for the third audio channel here.
ebnoThreshold	The Eb/No threshold value. The driver raises an alarm if the Eb/No ration falls below this value.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the IRD. The recommended setting is *19200,N,8,1* .

Preparing the IRD for remote control

Some settings at Scopus IRD's font panel must be made before the IRD can be connected to the M&C system. These settings are:

- Master-Slave protocol type
- Unit address 129 (recommended value)
- Remote interface RS232 (unless you are using the RS485 hardware).
- Remote interface baudrate 19200 (recommended value)
- C/Ku Band frequency display (Menue "CONFIG RECEIVER", see below).

DSNG frontend

The installed frontend of the Receiver is determined by the Fndt Command. If the system returns "S693" the IRD has an QPSK frontend and "D197" stand for DSNG frontend. The DSNG frontend supports some more functionality as the QPSK version.

L-band vs. RF frequency display

The Scopus IRD is capable to set/view the receive frequency either in terms of L-band or true

RF receive frequency. It however does not support the explicit programming of the LNB LO frequencies. To make the IRD accept and show true RF frequencies, you have to perform the following steps:

1. At the IRD front panel, go to the CONFIG RECEIVER menu and select the "satellite frequency" option which matches the LO frequencies your antenna uses.
2. At the sat-nms software, open the maintenance / setup dialog for the IRD and set the *loFreqHigh* , *loFreqLow* and *bandEdge* parameter to the proper frequencies. These values do not influence the receiver's display, but they are used by the driver to control the 22kHz tone while the *22Khz* parameter is set to *AUTO* .

If you want to display the receive frequency setting as a L-band value, set this at the receiver's front panel and the *loFreqHigh* , *loFreqLow* and *bandEdge* parameters to zero.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.frontend	TEXT	R/O	StringRange R/O
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (1 2)
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF AUTO)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 14V-Vertical 18V-Horizontal)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.audio.1	CHOICE	SAVE SETUP	EnumRange (OFF ANALOG DIGITAL)
config.audio.2	CHOICE	SAVE SETUP	EnumRange (OFF ANALOG DIGITAL)
config.audio.3	CHOICE	SAVE SETUP	EnumRange (OFF DIGITAL)
config.ebnoThreshold	FLOAT	SAVE SETUP	DoubleRange (1.0 .. 25.0)
internal.numServices	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ack	TEXT	R/O	StringRange R/O
internal.flags	HEX	R/O	HexRange R/O (0 .. 0)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mustBoot	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.bootTimer	TEXT	R/O	StringRange R/O
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.loFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fec	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange DIS (A B)
polarization	CHOICE		EnumRange (X Y)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
symbolRate	FLOAT		DoubleRange (1.00000 .. 44.00000)

name	*type*	*flags*	*range*
modulation	CHOICE		EnumRange DIS (QPSK 8PSK 16QAM AUTO)
rolloff	CHOICE		EnumRange DIS (35% 25% AUTO)
fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 AUTO)
programNo	INTEGER		IntegerRange (0 .. 0)
programList	TEXT	R/O	StringRange R/O
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.format	CHOICE		EnumRange (NTSC PAL BG PAL M PAL N SECAM PAL D)
video.aspect	CHOICE		EnumRange (4:3 16:9 AUTO OFF)
video.test	CHOICE		EnumRange (OFF NTSC PAL BG PAL M PAL N SECAM PAL D)
video.interpolation	CHOICE		EnumRange (Pass-through Pan-scan Letter-box)
audio.1.program	INTEGER		IntegerRange DIS (0 .. 99)
audio.1.routing	CHOICE		EnumRange DIS (OFF ON)
audio.1.output	CHOICE		EnumRange DIS (MONO STEREO)
audio.1.level	CHOICE		EnumRange DIS ()
audio.2.program	INTEGER		IntegerRange DIS (0 .. 99)
audio.2.routing	CHOICE		EnumRange DIS (OFF ON)
audio.2.output	CHOICE		EnumRange DIS (MONO STEREO)
audio.2.level	CHOICE		EnumRange DIS ()
audio.3.program	INTEGER		IntegerRange DIS (0 .. 99)
audio.3.routing	CHOICE		EnumRange DIS (OFF ON)
audio.3.output	CHOICE		EnumRange DIS (MONO STEREO)
audio.3.level	CHOICE		EnumRange DIS ()
audio.format	CHOICE		EnumRange (MPEG AC3 LINEAR)

name	*type*	*flags*	*range*
audio.ac3rate	CHOICE		EnumRange (32 kHz 44.1 kHz 48 kHz)
unitType	CHOICE		EnumRange (IRD DECODER)
decoderInput	CHOICE		EnumRange (SPI ASI RS422 INTERNAL)
digitalOutput	CHOICE		EnumRange (OFF ON)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mpeg	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ebno	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.signal	INTEGER	R/O	IntegerRange R/O (0 .. 100)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.aspect	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O
state.linkmargin	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.power	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.rsber	TEXT	R/O	StringRange R/O DIS
ca.bissmode	CHOICE		EnumRange (DSNG BISS-1 BISS-E-CLR BISS-E-INJ BISS-E-BUR)
ca.bisssession	TEXT		StringRange
ca.bissinjected	TEXT		StringRange
ca.dsngoddkey	TEXT		StringRange
ca.dsngevenkey	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Frontend not supported)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Signal lock)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (MPEG decode)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Eb/No)

9.620 Scopus-IRD2900-SNMP

This page describes the device driver and the device window for the Scopus 2900 IRD family, accessed by SNMP. Actually, the driver supports IRDs housing a single RF frontend and a single MPEG decoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the parameters concerning decoder input signal selection and the digital video output.
- --- Contains the conditional access parameters (BISS).
- --- Contains the parameters concerning the video output (Analog video format selection etc.)
- --- Contains the audio output parameters for all four audio outputs of the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Scopus IRD.

parameter	*description*
address	The driver expects the network address of the IRD in 'dotted quad' notation in this field. The address **must** be followed by the word 'private' in order to get configuration access to the IRD. Example:
'10.0.0.123 private' .	
22kHz	select if the IRD shall apply a 22kHz tone to the LNB.

parameter	***description***
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Select if the IRD shall apply LNB supply voltage. You may select 13V or 18V to make a LNB which supports this change the polarization
frequencyRange	Select the frequency band you want to use for frequency setting and display. If you select "L-BAND", the LO frequencies entered below are ignored and the frequency will be set as L-band value. The setting applies to the front panel operation as well.
loType	Select the type LNB / LO calculation to use. For normal applications you should set NUMERIC-Ku-BAND or NUMERIC-C-BAND. These settings allow the sat-nms device driver to do the frequency offset calculation.
loFreqHigh	Enter the LO frequency for the upper frequency band here.
loFreqLow	Enter the LO frequency for the lower frequency band here.
bandEdge	Enter frequency here, where the receiver shall switch to the upper frequency band.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

LO Frequency Calculation

The IRD itself does not provide a method to select two free definable LO frequencies which are selected by the desired input frequency. The sat-nms device driver works around this with the following trick:

With the configuration parameters *frequencyRange=Ku-BAND* and *loType=NUMERIC-Ku-Band* (or *C-Band* and *NUMERIC-C-BAND* respectively) are set, the device driver sets the applicable LO frequency at the IRD every time before the receive frequency is set. This tunes the receiver to the correct L-band frequency and displays the receive frequency in Ku-band or C-band as expected. Unlike the LO-frequency calculation built into the receiver, this also works with non-standard LO frequencies.

Remark: If you change the LO frequencies in the configuration dialog, this will not become effective until you set a new receive frequency at the IRDs main page.

DVB Mode Selection With RX-Channel-Select-CDB

The Scopus IRD2900 does not provide a separate parameter to set the DVB-S / DVB-S2 mode. It instead uses the *modulation* parameter to set the DVB mode and the modulation mode as a combined setting.

To make the sat-nms RX-ChannelSelect-CDB logical device set the DVB mode properly with a channel definition, the driver provides a parameter called *hidden.dvbMode* . This parameter sets the modulation setting to DVBS-QPSK for DVBS and to DVBS2-N-8PSK for DVBS2. When reading back this parameter, the driver interprets everything which is not DVBS-QPSK as DVBS2. This means, the DVBS/DVBS2 setting may be reported wrong if the setting has been made from another source than the RX-ChannelSelect-CDB device. As soon as there is another channel definition applied from the RX-ChannelSelect-CDB device, the parameter ist set correctly.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.dvbs2	TEXT	R/O	StringRange R/O
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 13V 18V)
config.frequencyRange	CHOICE	SETUP	EnumRange (L-BAND C-BAND Ku-BAND)
config.loType	CHOICE	SETUP	EnumRange (diSEqC UNIVERSAL WIDE-BAND NUMERIC-Ku-BAND NUMERIC-C-BAND)

name	*type*	*flags*	*range*
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
x.base	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.fend	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.mpeg	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vid1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.band	TEXT	R/O	StringRange R/O
internal.lbandfreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
input	CHOICE		EnumRange (1 2 3 4)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (1.000 .. 46.000)
modulation	CHOICE		EnumRange (QPSK)
fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 AUTO)
rolloff	CHOICE		EnumRange (25% 35% AUTO)
pilots	CHOICE		EnumRange (OFF ON)
hidden.dvbMode	CHOICE		EnumRange (DVBS DVBS2)

name	*type*	*flags*	*range*
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	TEXT	R/O	StringRange R/O
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
stream.inputRate	CHOICE		EnumRange (HIGH-SPEED FAST NORMAL)
stream.asiOutput	CHOICE		EnumRange (NONE DECRYPT-OFF DECRYPT-ON)
stream.27MHzClock	CHOICE		EnumRange (PCR INTERN EXTERN)
stream.decoderInput	CHOICE		EnumRange (NONE FRONT-END ASI RS422)
stream.type	CHOICE		EnumRange (ATSC DVB)
stream.asiMode	CHOICE		EnumRange (BURST NORMAL)
video.format	CHOICE		EnumRange (NTSC NTSC-PED-OFF PAL-M PAL-BG PAL-D PAL-N SECAM RU-SECAM)
video.aspect	CHOICE		EnumRange (TRANSP 16:9 16:9 LB 4:3 14:9 14:9 LB OFF)
video.fail	CHOICE		EnumRange (BLACK LAST-FLD LAST-FRM BAR NULL)
video.lipsync	CHOICE		EnumRange (STANDARD SINGLE ALWAYS OFF)
video.lsyncDelay	INTEGER		IntegerRange (0 .. 255)
video.interpolation	CHOICE		EnumRange (PASS-THROUGH PAN-SCAN LETTER-BOX)
audio.1.program	INTEGER		IntegerRange (0 .. 8191)
audio.1.level	INTEGER		IntegerRange (-64 .. 0)
audio.1.decode	CHOICE		EnumRange (MUSICAM DOLBY-AC3 AC3-PASSTHR LINEAR-PCM AUTO)
audio.1.routing	CHOICE		EnumRange (STEREO MONO SWAP RIGHT LEFT)
audio.1.output	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
audio.1.avsync	CHOICE		EnumRange (OFF ON)
audio.1.doutput	CHOICE		EnumRange (SPDIF AESEBU)
audio.1.ac3mix	CHOICE		EnumRange (DOLBY MONO STEREO CENTER MONO-SUR CENTER-MONO STEREO-SUR CENTER-STEREO)
audio.1.ac3mode	CHOICE		EnumRange (CUST-0 CUST-1 LINEOUT RF-REM)
audio.1.ac3srate	INTEGER		IntegerRange (0 .. 0)
audio.2.program	INTEGER		IntegerRange (0 .. 8191)
audio.2.level	INTEGER		IntegerRange (-64 .. 0)
audio.2.decode	CHOICE		EnumRange (MUSICAM DOLBY-AC3 AC3-PASSTHR LINEAR-PCM AUTO)
audio.2.routing	CHOICE		EnumRange (STEREO MONO SWAP RIGHT LEFT)
audio.2.output	CHOICE		EnumRange (OFF ON)
audio.2.avsync	CHOICE		EnumRange (OFF ON)
audio.2.doutput	CHOICE		EnumRange (SPDIF AESEBU)
audio.2.ac3mix	CHOICE		EnumRange (DOLBY MONO STEREO CENTER MONO-SUR CENTER-MONO STEREO-SUR CENTER-STEREO)
audio.2.ac3mode	CHOICE		EnumRange (CUST-0 CUST-1 LINEOUT RF-REM)
audio.2.ac3srate	INTEGER		IntegerRange (0 .. 0)
audio.3.program	INTEGER		IntegerRange (0 .. 8191)
audio.3.level	INTEGER		IntegerRange (-64 .. 0)
audio.3.decode	CHOICE		EnumRange (MUSICAM DOLBY-AC3 AC3-PASSTHR LINEAR-PCM AUTO)
audio.3.routing	CHOICE		EnumRange (STEREO MONO SWAP RIGHT LEFT)
audio.3.output	CHOICE		EnumRange (OFF ON)
audio.3.avsync	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
audio.3.doutput	CHOICE		EnumRange (SPDIF AESEBU)
audio.3.ac3mix	CHOICE		EnumRange (DOLBY MONO STEREO CENTER MONO-SUR CENTER-MONO STEREO-SUR CENTER-STEREO)
audio.3.ac3mode	CHOICE		EnumRange (CUST-0 CUST-1 LINEOUT RF-REM)
audio.3.ac3srate	INTEGER		IntegerRange (0 .. 0)
audio.4.program	INTEGER		IntegerRange (0 .. 8191)
audio.4.level	INTEGER		IntegerRange (-64 .. 0)
audio.4.decode	CHOICE		EnumRange (MUSICAM DOLBY-AC3 AC3-PASSTHR LINEAR-PCM AUTO)
audio.4.routing	CHOICE		EnumRange (STEREO MONO SWAP RIGHT LEFT)
audio.4.output	CHOICE		EnumRange (OFF ON)
audio.4.avsync	CHOICE		EnumRange (OFF ON)
audio.4.doutput	CHOICE		EnumRange (SPDIF AESEBU)
audio.4.ac3mix	CHOICE		EnumRange (DOLBY MONO STEREO CENTER MONO-SUR CENTER-MONO STEREO-SUR CENTER-STEREO)
audio.4.ac3mode	CHOICE		EnumRange (CUST-0 CUST-1 LINEOUT RF-REM)
audio.4.ac3srate	INTEGER		IntegerRange (0 .. 0)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mpeg	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.aspect	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.signal	INTEGER	R/O	IntegerRange R/O (0 .. 100)
ca.bissmode	CHOICE		EnumRange (DSNG BISS-1 BISS-E-CLR BISS-E-INJ BISS-E-BUR)
ca.bisssession	TEXT		StringRange
ca.dsngoddkey	TEXT		StringRange
ca.dsngevenkey	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (MPEG Lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (General)

9.621 Scopus-UE9000-SNMP

This page describes the device driver and the device window for the Scopus UE9000 MPEG video encoder family. The sat-nms device driver for the UE9000 encoders only implements a minimal set of parameters (mostly the modulator settings and a small number of encoding settings). The device window gives access to the encoder's web browser remote control through a button. A click to the button launches the web browser where you can control all settings of the encoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- Encoding / encryption parameters
- --- Device fault list
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The only parameter which must be set at the maintenance page of the device window is the encoder#s address:

parameter	*description*
address	Enter the IP address of the encoder here, followed by the community string 'private'. Example:

parameter	*description*
10.0.0.1 private	

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrelease	TEXT	R/O	StringRange R/O
info.fwrelease	TEXT	R/O	StringRange R/O
info.hwrelease	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.bucPower	CHOICE	SETUP	EnumRange (OFF ON)
config.bucReference	CHOICE	SETUP	EnumRange (OFF ON)
x.vid1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vid2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
x.aud3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.aud4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.asii	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vst1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vst2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.ast1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.ast2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.ast3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.ast4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vbi1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.vbi2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.asis	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.tstr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.lmod	INTEGER	R/O	IntegerRange R/O (0 .. 0)
x.srvc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.saveSettings	TEXT	R/O	StringRange R/O
internal.saveTimer	TEXT	R/O	StringRange R/O
internal.acodec.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.acodec.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.acodec.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.acodec.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.rfon	CHOICE	R/O	EnumRange R/O (OFF ON)
internal.type	TEXT	R/O	StringRange R/O
output.program	INTEGER		IntegerRange (1 .. 65535)
output.service	TEXT		StringRange
mod.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
mod.rfon	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
mod.power	FLOAT		DoubleRange (0 .. 0)
mod.modulation	CHOICE		EnumRange (OFF ON)
mod.type	CHOICE		EnumRange (DVB-S DVB-S2)
mod.fec	CHOICE		EnumRange (QPSK-NO-FEC QPSK-1/2 QPSK-2/3 QPSK-3/4 QPSK-5/6 QPSK-6/7 QPSK-7/8 QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-3/5 QPSK-4/5 QPSK-8/9 QPSK-9/10 APSK16-2/3 APSK16-3/4 APSK16-4/5 APSK16-5/6 APSK16-8/9 APSK16-9/10 16QAM-3/4 16QAM-7/8 8PSK-3/5 8PSK-2/3 8PSK-3/4 8PSK-5/6 8PSK-8/9 8PSK-9/10)
mod.modulation2	CHOICE		EnumRange (QPSK 16APSK 16QAM 8PSK)
mod.fec2	CHOICE		EnumRange (1/2 1/3 1/4 2/3 2/5 3/4 3/5 4/5 5/6 6/7 7/8 8/9 9/10 N/A)
mod.symbolrate	FLOAT		DoubleRange (0.00000 .. 0.00000)
mod.spectrum	CHOICE		EnumRange (NORM INVERT)
mod.rollOff	CHOICE		EnumRange (0.35 0.25 0.20)
mod.pilot	CHOICE		EnumRange (OFF ON)
video.aspect	CHOICE		EnumRange (4:3 16:9 4:3-AUTO 16:9-AUTO)
audio.1.avdelay	INTEGER		IntegerRange (-300 .. 300)
audio.2.avdelay	INTEGER		IntegerRange (-300 .. 300)
audio.3.avdelay	INTEGER		IntegerRange (-300 .. 300)
audio.4.avdelay	INTEGER		IntegerRange (-300 .. 300)
ca.type	CHOICE		EnumRange (CLEAR BISS-1 BISS-E-BURIED BISS-E-INJECTED)
ca.injected	TEXT		StringRange
ca.clearsw	TEXT		StringRange
ca.cryptedsw	TEXT		StringRange

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Encoder)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Audio 4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Modulator)
faultList	TEXT	R/O	StringRange R/O

9.622 Selector-Two-Way

The Selector-Two-Way logical device provides a simple way to show multiple parameter values in a single entry field of the user interface.

- The device selects one of up to 12 arbitrary parameters on the M&C and lets this parameter be monitored and controlled with its *value* parameter. The IDs of the parameters to select are defined in the device's setup with the *pos.##.id* configuration parameters.
- To select a parameter, the device listens to one parameter in the M&C and compares the actual reading of this parameter to the reference values defined in the device's setup with the *pos.##.select* configuration parameters. If the actual reading of the selection parameter matches the reference value of a position, this position gets selected.
- If the reading of the selection parameter matches none of the defined reference values or if the selection parameter provides no reading at all, a default position gets selected. The default position is defined in the device's setup as well.
- The parameter addressed by the selected position of the selector appears in the*value* parameter of the device. *value* always shows the actual value of this parameter.*value* also has the data type and range of the addressed parameter. This means you also may command the addressed parameter by changing *value* in the user interface. This makes the Selector-Two-Way work bidirectional, which gives the device its name.
- When selecting a parameter for *value*, the device also sets a parameter called *label* with a test defined for this selector position. You may use this *label* to identify in the user interface which parameter is actually seen in *value*.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- This page shows the selection state and actual value
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
selectId	The message ID for the parameter to monitor. e.g S1.position for a waveguide switch.
defaultPos	If you want to append a unit (or any other kind of text) to the output value, you can enter it here.
pos.01.id .. pos.12.id	The message ID of the parameter to monitor and control with the*value* parameter of this device if this particular position is selected.
pos.01.select .. pos.12.select	If the value of the parameter monitored with*selectId* equals the value entered here, the selector switches to this position. 'select' values must be unique, if two positions have the the same 'select' value configured, the device's behavior is not defined.
pos.01.label .. pos.12.label	The value to be set at the*label* parameter of this device if this particular position is selected.

Application example

You have to HPA's in an 1+1 redundant configuration. Both HPAs provides an output power, but you want only display the output power of the active HPA in the main screen to the operator. The waveguide switch's name is S1 with the position A and B and the HPA names are HPA-1 and HPA-2 :

parameter	***setting***
selsectId	S1.position
pos.01.id	HPA-1.meas.fwdPwr
pos.01.select	A
pos.01.label	HPA 1 power

parameter	*setting*
pos.02.id	HPA-2.meas.fwdPwr
pos.02.select	B
pos.02.label	HPA 2 power

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.selectPos	TEXT	R/O	StringRange R/O
info.selectVal	TEXT	R/O	StringRange R/O
config.selectId	TEXT	SAVE SETUP	StringRange
config.pos.01.id	TEXT	SAVE SETUP	StringRange
config.pos.01.select	TEXT	SAVE SETUP	StringRange
config.pos.01.label	TEXT	SAVE SETUP	StringRange
config.pos.02.id	TEXT	SAVE SETUP	StringRange
config.pos.02.select	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.pos.02.label	TEXT	SAVE SETUP	StringRange
config.pos.03.id	TEXT	SAVE SETUP	StringRange
config.pos.03.select	TEXT	SAVE SETUP	StringRange
config.pos.03.label	TEXT	SAVE SETUP	StringRange
config.pos.04.id	TEXT	SAVE SETUP	StringRange
config.pos.04.select	TEXT	SAVE SETUP	StringRange
config.pos.04.label	TEXT	SAVE SETUP	StringRange
config.pos.05.id	TEXT	SAVE SETUP	StringRange
config.pos.05.select	TEXT	SAVE SETUP	StringRange
config.pos.05.label	TEXT	SAVE SETUP	StringRange
config.pos.06.id	TEXT	SAVE SETUP	StringRange
config.pos.06.select	TEXT	SAVE SETUP	StringRange
config.pos.06.label	TEXT	SAVE SETUP	StringRange
config.pos.07.id	TEXT	SAVE SETUP	StringRange
config.pos.07.select	TEXT	SAVE SETUP	StringRange
config.pos.07.label	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.pos.08.id	TEXT	SAVE SETUP	StringRange
config.pos.08.select	TEXT	SAVE SETUP	StringRange
config.pos.08.label	TEXT	SAVE SETUP	StringRange
config.pos.09.id	TEXT	SAVE SETUP	StringRange
config.pos.09.select	TEXT	SAVE SETUP	StringRange
config.pos.09.label	TEXT	SAVE SETUP	StringRange
config.pos.10.id	TEXT	SAVE SETUP	StringRange
config.pos.10.select	TEXT	SAVE SETUP	StringRange
config.pos.10.label	TEXT	SAVE SETUP	StringRange
config.pos.11.id	TEXT	SAVE SETUP	StringRange
config.pos.11.select	TEXT	SAVE SETUP	StringRange
config.pos.11.label	TEXT	SAVE SETUP	StringRange
config.pos.12.id	TEXT	SAVE SETUP	StringRange
config.pos.12.select	TEXT	SAVE SETUP	StringRange
config.pos.12.label	TEXT	SAVE SETUP	StringRange
config.defaultPos	CHOICE	SAVE SETUP	EnumRange (01 02 03 04 05 06 07 08 09 10 11 12)
value	TEXT		StringRange

name	*type*	*flags*	*range*
label	TEXT	R/O	StringRange R/O

9.623 Selector

The Selector logical device provides a simple mechanism to monitor a switch position and up to 8 input values and set an output value regarding this switch position.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the selection state
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	*description*
switchId	The message ID for the switch to monitor. e.g S1.position for a waveguide switch.
unit	If you want to append a unit (or any other kind of text) to the output value, you can enter it here.
inputId.01	The message ID for the first input
inputId.02	The message ID for the second input
inputId.03	The message ID for input 3, leave empty if not needed
inputId.04	The message ID for input 4, leave empty if not needed
inputId.05	The message ID for input 5, leave empty if not needed
inputId.06	The message ID for input 6, leave empty if not needed
inputId.07	The message ID for input 7, leave empty if not needed
inputId.08	The message ID for input 8, leave empty if not needed
location.01	The value of the switchId parameter which represents the first input

parameter	*description*
location.02	The value of the switchId parameter which represents the second input
location.03	The value of the switchId parameter which represents the input 3, leave default (003) if not needed
location.04	The value of the switchId parameter which represents the input 4, leave default (004) if not needed
location.05	The value of the switchId parameter which represents the input 5, leave default (005) if not needed
location.06	The value of the switchId parameter which represents the input 6, leave default (006) if not needed
location.07	The value of the switchId parameter which represents the input 7, leave default (007) if not needed
location.08	The value of the switchId parameter which represents the input 8, leave default (008) if not needed

Application example

You have to HPA's in an 1+1 redundant configuration. Both HPAs provides an output power, but you want only display the output power of the active HPA in the main screen to the operator. The waveguide switch's name is S1 with the position A and B and the HPA names are HPA-1 and HPA-2 :

parameter	*setting*
switchId	S1.position
unit	dBm
inputId.01	HPA-1.meas.fwdPwr
inputId.02	HPA-2.meas.fwdPwr
location.01	A
location.02	B

Use [device-name].position to show the output power of the active HPA.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.inputId.01	TEXT	SAVE SETUP	StringRange
config.location.01	TEXT	SAVE SETUP	StringRange
config.preConst.01	TEXT	SAVE SETUP	StringRange
config.postConst.01	TEXT	SAVE SETUP	StringRange
config.inputId.02	TEXT	SAVE SETUP	StringRange
config.location.02	TEXT	SAVE SETUP	StringRange
config.preConst.02	TEXT	SAVE SETUP	StringRange
config.postConst.02	TEXT	SAVE SETUP	StringRange
config.inputId.03	TEXT	SAVE SETUP	StringRange
config.location.03	TEXT	SAVE SETUP	StringRange
config.preConst.03	TEXT	SAVE SETUP	StringRange
config.postConst.03	TEXT	SAVE SETUP	StringRange
config.inputId.04	TEXT	SAVE SETUP	StringRange
config.location.04	TEXT	SAVE SETUP	StringRange
config.preConst.04	TEXT	SAVE SETUP	StringRange
config.postConst.04	TEXT	SAVE SETUP	StringRange
config.inputId.05	TEXT	SAVE SETUP	StringRange
config.location.05	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.preConst.05	TEXT	SAVE SETUP	StringRange
config.postConst.05	TEXT	SAVE SETUP	StringRange
config.inputId.06	TEXT	SAVE SETUP	StringRange
config.location.06	TEXT	SAVE SETUP	StringRange
config.preConst.06	TEXT	SAVE SETUP	StringRange
config.postConst.06	TEXT	SAVE SETUP	StringRange
config.inputId.07	TEXT	SAVE SETUP	StringRange
config.location.07	TEXT	SAVE SETUP	StringRange
config.preConst.07	TEXT	SAVE SETUP	StringRange
config.postConst.07	TEXT	SAVE SETUP	StringRange
config.inputId.08	TEXT	SAVE SETUP	StringRange
config.location.08	TEXT	SAVE SETUP	StringRange
config.preConst.08	TEXT	SAVE SETUP	StringRange
config.postConst.08	TEXT	SAVE SETUP	StringRange
config.unit	TEXT	SAVE SETUP	StringRange
config.default	TEXT	SAVE SETUP	StringRange
position	TEXT		StringRange
output	TEXT		StringRange
value	FLOAT		DoubleRange (0.00 .. 0.00)
input.01	TEXT		StringRange
input.02	TEXT		StringRange
input.03	TEXT		StringRange
input.04	TEXT		StringRange
input.05	TEXT		StringRange
input.06	TEXT		StringRange
input.07	TEXT		StringRange

name	*type*	*flags*	*range*
input.08	TEXT		StringRange
internal.output	TEXT		StringRange

9.624 Sencore-AG4400

This page describes the device driver and the device window for one Sencore AG440 receiver unit in an OpenFrame. Each AG4400 in the OpenFrame is controlled as an individual device with it's own IP address.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the parameters concerning the video, video test output and embedded audio assignment.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read

and write community.

- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

LO frequency translation is done by the Sencore-AG4400 driver, not by the device itself. Hence, the LO frequency **must** be set to zero at the receiver's Web-GUI for both inputs. The sat-nms device driver does not know which LO frequency to use at system startup. For this reason the sat-nms GUI shows the actual L-band receive frequency after system start, as long as no receive frequency has been set by the operator.

The input redundancy switching functions of the IRD are not used with this driver. The driver switches off this function every time, a tuner parameter (frequency, symbol rate or input selection) is set. This is to provide compatibility of the input selection behavior with other IRDs supported by the sat-nms software.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP NOCOMPARE	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 13V 14V 18V 19V)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input	CHOICE		EnumRange (A B ASI)
dvbmode	CHOICE		EnumRange (DVBS DVBS2 AUTO)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRateMode	CHOICE		EnumRange (AUTO MANUAL)
modulation	CHOICE	R/O	EnumRange R/O (n/a QPSK 8PSK 16APSK 32APSK)

name	*type*	*flags*	*range*
fec	CHOICE	R/O	EnumRange R/O (?/? 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 6/7 7/8 8/9 9/10 n/a)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.format	TEXT	R/O	StringRange R/O
video.chroma	TEXT	R/O	StringRange R/O
video.rasterMode	CHOICE		EnumRange (SOLID COLOR LAST FRAME)
video.rasterColor	CHOICE		EnumRange (BLACK WHITE YELLOW CYAN MAGENTA RED BLUE GREEN GRAY)
video.hdDisplayMode	CHOICE		EnumRange (CENTERCUT PILLARBOX ANAMORPHIC)
video.hdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.hdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.sdDisplayMode	CHOICE		EnumRange (CENTERCUT LETTERBOX ANAMORPHIC)
video.sdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.sdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.formatSelect	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
video.manualFormat	CHOICE		EnumRange (720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9))
video.outputFormat	CHOICE	R/O	EnumRange R/O (UNKNOWN 720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9))
video.embeddedAudio1	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)
video.embeddedAudio2	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)

name	*type*	*flags*	*range*
video.embeddedAudio3	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)
video.embeddedAudio4	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.samplingRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.codingMode	TEXT	R/O	StringRange R/O
audio.1.formatMode	CHOICE		EnumRange (CONSUMER PROFESSIONAL)
audio.1.downmix	CHOICE		EnumRange (LORO-STEREO LTRT-DOLBY-SURROUND LTRT-AUTO DUAL-MONO DUAL-LEFT DUAL-RIGHT)
audio.1.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.1.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.samplingRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.codingMode	TEXT	R/O	StringRange R/O
audio.2.formatMode	CHOICE		EnumRange (CONSUMER PROFESSIONAL)

name	*type*	*flags*	*range*
audio.2.downmix	CHOICE		EnumRange (LORO-STEREO LTRT-DOLBY-SURROUND LTRT-AUTO DUAL-MONO DUAL-LEFT DUAL-RIGHT)
audio.2.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.2.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.3.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.samplingRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.codingMode	TEXT	R/O	StringRange R/O
audio.3.formatMode	CHOICE		EnumRange (CONSUMER PROFESSIONAL)
audio.3.downmix	CHOICE		EnumRange (LORO-STEREO LTRT-DOLBY-SURROUND LTRT-AUTO DUAL-MONO DUAL-LEFT DUAL-RIGHT)
audio.3.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.3.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.4.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.samplingRate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.codingMode	TEXT	R/O	StringRange R/O
audio.4.formatMode	CHOICE		EnumRange (CONSUMER PROFESSIONAL)
audio.4.downmix	CHOICE		EnumRange (LORO-STEREO LTRT-DOLBY-SURROUND LTRT-AUTO DUAL-MONO DUAL-LEFT DUAL-RIGHT)

name	*type*	*flags*	*range*
audio.4.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.4.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.aes.1.output	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)
audio.aes.2.output	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH AUDIO 3 PCM AUDIO 3 PASS THROUGH AUDIO 4 PCM AUDIO 4 PASS THROUGH)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.carrier	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
state.dvbmode	CHOICE	R/O	EnumRange R/O (DVBS DVBS2 n/a)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.pilots	CHOICE	R/O	EnumRange R/O (ON OFF)
state.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Carrier lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sync lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TS present)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (TS error)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Video)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.e	TEXT	R/O	StringRange R/O
internal.band	CHOICE	R/O	EnumRange R/O (UNKNOWN LO HI)
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

9.625 Sencore-Afiniti-Encoder

This page describes the device driver and the device window for the Sencore Afiniti Encoder device via SNMP. It supports RTP/UDP and SRT output.

When using UHD (channel.mode = SINGLE) encoders instances 2-4 are readonly.

Driver support one HD encoding instance. For each additional instance another device is needed. Additionally it is necessary to select the slot where encoding card is installed via device setup config.slot. The dedicated Output can be selected via config.output

The driver was tested with the following sw releases:

- 5.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the MPEG output parameter settings and the operation control.
- --- Contains RTP/UDP and SRT output parameters and statistics.

- --- Contains the conditional access (BISS) parameters.
- --- Contains video in/output parameters from the MPEG stream.
- --- Contains the parameters concerning the audio channel 1 encoding.
- --- Contains the parameters concerning the audio channel 2 encoding.
- --- Contains the parameters concerning the audio channel 3 encoding.
- --- Contains the parameters concerning the audio channel 4 encoding.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- slot --- Defines the slot where encoding card is installed 1-6.
- channel --- Defines the used encoder for this device from the 1-4.
- output --- Defines the index of the ts output which should be allocated to the chosen encoding channel.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.mainModel	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.mainVersion	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.mainSerial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.slot	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.channel	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
config.outputType	CHOICE	SAVE SETUP	EnumRange (ASI IP)
config.output	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
internal.channel	TEXT	R/O	StringRange R/O
internal.output	TEXT	R/O	StringRange R/O
internal.channelMode	CHOICE	R/O	EnumRange R/O (SINGLE MULTI)
internal.bissmode	TEXT	R/O	StringRange R/O
internal.encryption	CHOICE	R/O	EnumRange R/O (ON OFF)
internal.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hSize	TEXT	R/O	StringRange R/O
internal.vSize	TEXT	R/O	StringRange R/O
internal.frame	TEXT	R/O	StringRange R/O
internal.prog	TEXT	R/O	StringRange R/O
internal.audio1	TEXT	R/O	StringRange R/O
internal.audio2	TEXT	R/O	StringRange R/O
internal.audio3	TEXT	R/O	StringRange R/O
internal.audio4	TEXT	R/O	StringRange R/O
internal.protocol	TEXT	R/O	StringRange R/O
internal.versMajorCompare	TEXT	R/O	StringRange R/O
internal.versMinorCompare	TEXT	R/O	StringRange R/O
internal.versionIsMin5.4	INTEGER	R/O	IntegerRange R/O (0 .. 1)
video.codec	CHOICE		EnumRange (MPEG2 MPEG4-AVC HEVC)
channel.mode	CHOICE		EnumRange (SINGLE MULTI)
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.source	TEXT	R/O	StringRange R/O
output.latency	CHOICE		EnumRange (NORMAL LOW VERY-LOW LONG)
output.latencyTrim	INTEGER		IntegerRange (-400 .. 100)
output.datarate	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
output.program	INTEGER		IntegerRange (1 .. 65535)
output.programName	TEXT		StringRange
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.tsid	INTEGER		IntegerRange (0 .. 65535)
output.nitpid	INTEGER		IntegerRange (0 .. 65535)
output.pmtpid	INTEGER		IntegerRange (0 .. 8191)
output.muxrate	FLOAT	R/O	DoubleRange R/O (1.000 .. 210.000)
video.pid	INTEGER		IntegerRange (0 .. 8190)
video.startup	CHOICE		EnumRange (ON OFF)
video.chroma	CHOICE		EnumRange (4:2:0 4:2:2)
video.bitdepth	CHOICE		EnumRange (8-Bit 10-Bit)
video.aspectRatio	CHOICE		EnumRange (4:3 16:9)
video.gopType	CHOICE		EnumRange (CLOSED OPEN)
video.gopStruct	CHOICE		EnumRange (IP IBP IBBP)
video.gopLength	INTEGER		IntegerRange (1 .. 64)
video.autoFill	CHOICE		EnumRange (OFF ON)
video.loss	CHOICE		EnumRange (STOP ENCODING BARS BARS+ID)
video.lossFormat	CHOICE		EnumRange (480i59.94 576i50 720p59.94 720p50 1080i59.94 1080i50 1080p59.94 1080p50)
video.datarate	FLOAT		DoubleRange (1.000 .. 94.916)
video.uhdMode	CHOICE		EnumRange (OFF 2-SI SQUARE)
video.uhdlossFormat	CHOICE		EnumRange (2160p59.94 2160p50 2160p29.97 2160p25)
ca.bissmode	CHOICE		EnumRange (OFF BISS-1 BISS-E)

name	*type*	*flags*	*range*
ca.bisskey	TEXT	NOPRESET	StringRange
ca.injectedid	TEXT	NOPRESET	StringRange
ipout.1.enable	CHOICE		EnumRange (ON OFF)
ipout.1.interface	CHOICE		EnumRange (PORT-1 PORT-2)
ipout.1.datarate	FLOAT		DoubleRange (1.000 .. 210.000)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.protocol	CHOICE		EnumRange (RTP UDP SRT)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.buffer	INTEGER		IntegerRange (0 .. 500)
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.latency	INTEGER		IntegerRange (0 .. 60000)
srtout.1.packetDelivery	CHOICE		EnumRange (OFF ON DEFAULT)
srtout.1.peerLatency	INTEGER		IntegerRange (0 .. 60000)
srtout.1.encryption	CHOICE		EnumRange (OFF AES-128 AES-192 AES-256)
srtout.1.key	TEXT	NOPRESET	StringRange
srtout.1.state	TEXT	R/O	StringRange R/O
srtout.1.rtt	TEXT	R/O	StringRange R/O
srtout.1.datarate	TEXT	R/O	StringRange R/O
srtout.1.packetsRetr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
srtout.1.buffer	TEXT	R/O	StringRange R/O
audio.1.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)

name	*type*	*flags*	*range*
audio.1.delay	INTEGER		IntegerRange (-800 .. 800)
audio.1.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.1.profile	CHOICE		EnumRange (MPEG1-L2 03 AAC-LC)
audio.1.language	TEXT		StringRange
audio.1.level	INTEGER		IntegerRange (-18 .. 8)
audio.2.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.delay	INTEGER		IntegerRange (-800 .. 800)
audio.2.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.2.profile	CHOICE		EnumRange (MPEG1-L2 03 AAC-LC)
audio.2.language	TEXT		StringRange
audio.2.level	INTEGER		IntegerRange (-18 .. 8)
audio.3.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.delay	INTEGER		IntegerRange (-800 .. 800)

name	*type*	*flags*	*range*
audio.3.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.3.profile	CHOICE		EnumRange (MPEG1-L2 03 AAC-LC)
audio.3.language	TEXT		StringRange
audio.3.level	INTEGER		IntegerRange (-18 .. 8)
audio.4.mode	CHOICE		EnumRange (OFF ENCODE PASSTHROUGH)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.delay	INTEGER		IntegerRange (-800 .. 800)
audio.4.datarate	CHOICE		EnumRange (48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 160kbps 192kbps 224kbps 256kbps 320kbps 384kbps 448kbps 512kbps 576kbps 640kbps 1920kbps 2308kbps)
audio.4.profile	CHOICE		EnumRange (MPEG1-L2 03 AAC-LC)
audio.4.language	TEXT		StringRange
audio.4.level	INTEGER		IntegerRange (-18 .. 8)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)

9.626 Sencore-MIP6210

This page describes the device driver and the device window for the Sencore MIP6210 Stream Recorder and Player, controlled via SNMP.

The driver was tested with sw release**1.10.0**

The driver can only handle one recording and one player instance. As the MIP6210 can handle various instances, each instance has to be setup in a single device. The respective instance has

to be configured after first initialization via config.recorder and config.player which are read from the available instances on the unit.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the recorder input parameters and file statistics.
- --- Contains the player output parameters and file statistics.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the MIP6210.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- recorder --- Defines the desired recorder instance.
- player --- Defines the desired player instance.
- cpuAlarm --- Defines maximum cpu usage of the unit. If the cpu usage exceeds the defined threshold a CPU Usage fault is generated.
- deviceComment --- An optional text entered here appears as an addition to the tooltip

which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.uuid	TEXT	R/O	StringRange R/O
info.cpuUsage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.recorder	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8)
config.player	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6 7 8)
config.cpuAlarm	INTEGER	SAVE SETUP	IntegerRange (0 .. 100)
config.deviceComment	TEXT	SAVE SETUP	StringRange
recorderId	TEXT	R/O	StringRange R/O
playerId	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
fileId	TEXT	R/O	StringRange R/O
msecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
msecs2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
msecs3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
msecs4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
seconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)
minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
timestring	TEXT	R/O	StringRange R/O
internal.filesize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.file	CHOICE	R/O	EnumRange R/O (unknown)
ipin.interface	CHOICE	R/O	EnumRange R/O (ETH-0 ETH-1)
ipin.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.address	TEXT		StringRange
ipin.port	INTEGER		IntegerRange (0 .. 65535)
ipin.igmpFilter	CHOICE		EnumRange (INCLUDE EXCLUDE)
ipin.maxDuration.hr	INTEGER		IntegerRange (0 .. 0)
ipin.maxDuration.mi	INTEGER		IntegerRange (0 .. 60)
ipin.maxDuration.se	INTEGER		IntegerRange (0 .. 60)
ipin.record	CHOICE	NOPRESET	EnumRange ()
ipin.stop	TEXT	NOPRESET	StringRange
ipin.alias	TEXT	R/O	StringRange R/O
ipin.duration	TEXT	R/O	StringRange R/O
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O (1 .. 7)
ipin.protocol	CHOICE	R/O	EnumRange R/O (UDP RTP)
ipin.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
ipin.recordState	CHOICE	R/O	EnumRange R/O (RECORDING DELAY STOPPED)
ipin.tsState	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.linkState	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.state	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.comment	TEXT	SAVE	StringRange
ipout.file	CHOICE	R/O	EnumRange R/O (unknown)
ipout.interface	CHOICE	R/O	EnumRange R/O (ETH-0 ETH-1)
ipout.address	TEXT		StringRange
ipout.port	INTEGER		IntegerRange (0 .. 65535)
ipout.protocol	CHOICE		EnumRange (UDP RTP)
ipout.srcMode	CHOICE		EnumRange (AUTO MANUAL)
ipout.srcAddress	TEXT		StringRange
ipout.srcPort	INTEGER		IntegerRange (0 .. 65535)
ipout.packetsFrame	INTEGER		IntegerRange (1 .. 7)
ipout.alias	TEXT		StringRange
ipout.bitrateMode	CHOICE		EnumRange (AUTO MANUAL)
ipout.datarate	FLOAT		DoubleRange (0.000000 .. 0.000000)
ipout.loop	CHOICE		EnumRange (ON OFF)
ipout.seamlessLoop	CHOICE		EnumRange (ON OFF)
ipout.comment	TEXT	SAVE	StringRange
ipout.play	TEXT	NOPRESET	StringRange
ipout.pause	TEXT	NOPRESET	StringRange
ipout.stop	TEXT	NOPRESET	StringRange
ipout.mode	CHOICE	R/O	EnumRange R/O (MULTICAST UNICAST)

name	*type*	*flags*	*range*
ipout.measDatarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipout.playState	CHOICE	R/O	EnumRange R/O (UNKNOWN PLAYING PAUSED STOPPED)
ipout.receiverMac	TEXT	R/O	StringRange R/O
ipout.filesize	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
ipout.progress	TEXT	R/O	StringRange R/O
ipout.duration	TEXT	R/O	StringRange R/O
ipout.useStartEnd	CHOICE		EnumRange (ON OFF)
ipout.startTime.hr	INTEGER		IntegerRange (0 .. 0)
ipout.startTime.mi	INTEGER		IntegerRange (0 .. 60)
ipout.startTime.se	INTEGER		IntegerRange (0 .. 60)
ipout.endTime.hr	INTEGER		IntegerRange (0 .. 0)
ipout.endTime.mi	INTEGER		IntegerRange (0 .. 60)
ipout.endTime.se	INTEGER		IntegerRange (0 .. 60)
flags.record	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.play	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Record TS Sync)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Record Link)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Play Unicast Rec)
faults.04	ALARM	R/O	AlarmFlagRange R/O (CPU Usage)

9.627 Sencore-MRD3187-DVBS2

This page describes the device driver and the device window for one DVB-S2 receiver unit in a Sencore MRD3187 device. The Sencore MRD3187 may contain up to two receiver / decoder boards, this driver controls one DVB-S2 receiver board in the MRD. To control two boards in one unit, set up two sat-nms devices with the same IP address, but different 'rds' configuration

settings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video, video test output and embedded audio assignment.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the MRD3187.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- rds --- Addresses the RDS DVB-S2 receiver in the MRD unit (1/2). To control both IRDs in one unit, set up two sat-nms devices which use the same IP address, nut one configured to rds=1, the to rds=2.
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- ebnoMin --- This value is used to compute in the driver a link margin from the Eb/No reading provided by the device. The displayed margin value is (Eb/No - ebnoMin).
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The Sencore MRD3187 provides no means to translate frequency settings with a LO frequency. Hence, the frequency translation has been entirely implemented in the driver: At the sat-nms GUI the receive frequency is set as "true RX frequency", the Sencore WebGUI shows the L-band receive frequency.

As the LO frequency selection is not coupled to a switch or LNB supply setting, the sat-nms device driver does not know which LO frequency to use at system startup. For this reason the sat-nms GUI shows the actual L-band receive frequency after system start, as long as not receive frequency has been set by the operator.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serialno	TEXT	R/O	StringRange R/O
info.location	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.rds	CHOICE	SAVE SETUP	EnumRange (1 2)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ebnoMin	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input	CHOICE		EnumRange (A B)
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE	R/O	EnumRange R/O (???? QPSK 8PSK)
fec	CHOICE	R/O	EnumRange R/O (?/? 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 6/7 7/8 8/9 9/10)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
actualProgram	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.format	TEXT	R/O	StringRange R/O
video.chroma	TEXT	R/O	StringRange R/O
video.sdiA	CHOICE		EnumRange (AUTO SD HD)
video.sdiB	CHOICE		EnumRange (AUTO SD HD)
video.hdDisplayMode	CHOICE		EnumRange (PILLARBARS CROPPED)
video.hdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.hdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.sdDisplayMode	CHOICE		EnumRange (LETTERBOX CROPPED ANAMORPHIC)
video.sdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.sdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.formatSelect	CHOICE		EnumRange (AUTO MANUAL)
video.manualFormat	CHOICE		EnumRange (720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9) 1920x1080 @ 23.976psf (16:9) 1920x1080 @ 24psf (16:9))

name	*type*	*flags*	*range*
video.outputFormat	CHOICE	R/O	EnumRange R/O (UNKNOWN 720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9) 1920x1080 @ 23.976psf (16:9) 1920x1080 @ 24psf (16:9))
video.embeddedAudio1	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
video.embeddedAudio2	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
video.embeddedAudio3	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
video.embeddedAudio4	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.samplingRate	CHOICE	R/O	EnumRange R/O (UNKNOWN 48 kHz 44.1 kHz 32 kHz RESERVED)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.codingMode	TEXT	R/O	StringRange R/O
audio.1.decoderType	CHOICE		EnumRange (MONITOR TRANSMISSION USER)
audio.1.downmix	CHOICE		EnumRange (2/0 L/R 2/0 LT/RT MONO CHAN1 MONO CHAN2 2/0 AUTO)
audio.1.dynamic	CHOICE		EnumRange (ENABLED DISABLED)

name	*type*	*flags*	*range*
audio.1.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.1.compressMon	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.samplingRate	CHOICE	R/O	EnumRange R/O (UNKNOWN 48 kHz 44.1 kHz 32 kHz RESERVED)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.codingMode	TEXT	R/O	StringRange R/O
audio.2.decoderType	CHOICE		EnumRange (MONITOR TRANSMISSION USER)
audio.2.downmix	CHOICE		EnumRange (2/0 L/R 2/0 LT/RT MONO CHAN1 MONO CHAN2 2/0 AUTO)
audio.2.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.2.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.2.compressMon	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.carrier	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.pilots	CHOICE	R/O	EnumRange R/O (?? ON OFF)
state.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ca.bissmode	CHOICE		EnumRange (BISS-1 BISS-E)

name	*type*	*flags*	*range*
ca.bisskey	TEXT		StringRange
ca.cam1enable	CHOICE		EnumRange (DISABLE ENABLE)
ca.cam1type	TEXT	R/O	StringRange R/O
ca.cam2enable	CHOICE		EnumRange (DISABLE ENABLE)
ca.cam2type	TEXT	R/O	StringRange R/O
ca.casmode	CHOICE		EnumRange (AUTO MANUAL)
ca.casid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ca.casname	TEXT	R/O	StringRange R/O
ca.casmonitor	CHOICE		EnumRange (DISABLE ENABLE)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Carrier lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sync lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TS present)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TS error)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
internal.band	CHOICE	R/O	EnumRange R/O (UNKNOWN LO HI)
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)

9.628 Sencore-MRD3187-QUAD

This page describes the device driver and the device window for one Quad-input DVB-S2 receiver unit in a Sencore MRD3187 device. The Sencore MRD3187 may contain up to two receiver / decoder boards, this driver controls one 4-input receiver board in the MRD. To control two boards in one unit, set up two sat-nms devices with the same IP address, but different 'rds' configuration settings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video, video test output and embedded audio assignment.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the MRD3187.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- rds --- Addresses the RDS receiver in the MRD unit (1/2). To control both IRDs in one unit, set up two sat-nms devices which use the same IP address, nut one configured to rds=1, the to rds=2.
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- ebnoMin --- This value is used to compute in the driver a link margin from the Eb/No

reading provided by the device. The displayed margin value is (Eb/No - ebnoMin).

- lnbPowerA --- The LNB power setting for input A
- lnbPowerB --- The LNB power setting for input B
- lnbPowerC --- The LNB power setting for input C
- lnbPowerD --- The LNB power setting for input D
- lnb22kHzA --- The 22 kHz tone setting for input A
- lnb22kHzB --- The 22 kHz tone setting for input B
- lnb22kHzC --- The 22 kHz tone setting for input C
- lnb22kHzD --- The 22 kHz tone setting for input D
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The QUAD DVB-S/DVB-S2 stores certain parameters in individual memories for each RF input. This applies to the receive frequency, the LO frequency, the symbol rate and the DVB-S/DVB-S2 setting. The sat-nms software always shows the values for the actually selected input. When writing one of these parameters, the software sets the new value for all four inputs.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.alias	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.location	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.rds	CHOICE	SAVE SETUP	EnumRange (1 2)
config.hasDualIp	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasQuadDVB	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasAsi	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.ebnoMin	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.lnbPowerA	CHOICE	SETUP	EnumRange (OFF 13V 14V 18V 19V)
config.lnbPowerB	CHOICE	SETUP	EnumRange (OFF 13V 14V 18V 19V)
config.lnbPowerC	CHOICE	SETUP	EnumRange (OFF 13V 14V 18V 19V)
config.lnbPowerD	CHOICE	SETUP	EnumRange (OFF 13V 14V 18V 19V)
config.lnb22kHzA	CHOICE	SETUP	EnumRange (OFF ON)
config.lnb22kHzB	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.lnb22kHzC	CHOICE	SETUP	EnumRange (OFF ON)
config.lnb22kHzD	CHOICE	SETUP	EnumRange (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input	CHOICE		EnumRange (A B C D)
interface	CHOICE		EnumRange (ASI IP SAT)
asi.type	CHOICE		EnumRange (ASI 310M ASI NULL STRIPPED UNKNOWN)
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE	R/O	EnumRange R/O (???? QPSK 8PSK)
fec	CHOICE	R/O	EnumRange R/O (?/? 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 6/7 7/8 8/9 9/10)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.format	TEXT	R/O	StringRange R/O
video.chroma	TEXT	R/O	StringRange R/O
video.sdiA	CHOICE		EnumRange (AUTO SD HD)
video.sdiB	CHOICE		EnumRange (AUTO SD HD)
video.hdDisplayMode	CHOICE		EnumRange (PILLARBARS CROPPED)

name	*type*	*flags*	*range*
video.hdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.hdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.sdDisplayMode	CHOICE		EnumRange (LETTERBOX CROPPED ANAMORPHIC)
video.sdVideoLoss	CHOICE		EnumRange (NO OUTPUT DISP RASTER)
video.sdAutoAfd	CHOICE		EnumRange (ENABLED DISABLED)
video.formatSelect	CHOICE		EnumRange (AUTO MANUAL)
video.manualFormat	CHOICE		EnumRange (720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9) 1920x1080 @ 23.976psf (16:9) 1920x1080 @ 24psf (16:9))
video.outputFormat	CHOICE	R/O	EnumRange R/O (UNKNOWN 720x480 @ 29.97i (4:3) 720x480 @ 29.97i (16:9) 720x576 @ 25i (4:3) 720x576 @ 25i (16:9) 1280x720 @ 50p (16:9) 1280x720 @ 59.94p (16:9) 1280x720 @ 60p (16:9) 1920x1080 @ 25i (16:9) 1920x1080 @ 29.97i (16:9) 1920x1080 @ 30i (16:9) 1920x1080 @ 23.976p (16:9) 1920x1080 @ 24p (16:9) 1920x1080 @ 25p (16:9) 1920x1080 @ 29.97p (16:9) 1920x1080 @ 30p (16:9) 1920x1080 @ 23.976psf (16:9) 1920x1080 @ 24psf (16:9))
video.embeddedAudio1	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
video.embeddedAudio2	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)

name	*type*	*flags*	*range*
video.embeddedAudio3	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
video.embeddedAudio4	CHOICE		EnumRange (OFF AUDIO 1 PCM AUDIO 1 PASS THROUGH AUDIO 2 PCM AUDIO 2 PASS THROUGH)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.samplingRate	CHOICE	R/O	EnumRange R/O (UNKNOWN 48 kHz 44.1 kHz 32 kHz RESERVED)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.codingMode	TEXT	R/O	StringRange R/O
audio.1.decoderType	CHOICE		EnumRange (MONITOR TRANSMISSION USER)
audio.1.downmix	CHOICE		EnumRange (2/0 L/R 2/0 LT/RT MONO CHAN1 MONO CHAN2 2/0 AUTO)
audio.1.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.1.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.1.compressMon	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.samplingRate	CHOICE	R/O	EnumRange R/O (UNKNOWN 48 kHz 44.1 kHz 32 kHz RESERVED)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.codingMode	TEXT	R/O	StringRange R/O
audio.2.decoderType	CHOICE		EnumRange (MONITOR TRANSMISSION USER)
audio.2.downmix	CHOICE		EnumRange (2/0 L/R 2/0 LT/RT MONO CHAN1 MONO CHAN2 2/0 AUTO)
audio.2.dynamic	CHOICE		EnumRange (ENABLED DISABLED)
audio.2.compression	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)

name	*type*	*flags*	*range*
audio.2.compressMon	CHOICE		EnumRange (RF MODE LINE MODE CUSTOM 0 CUSTOM 1)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.carrier	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts.dvbs	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts.ip1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts.ip2	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.pilots	CHOICE	R/O	EnumRange R/O (?? ON OFF)
state.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ca.bissmode	CHOICE		EnumRange (BISS-1 BISS-E)
ca.bisskey	TEXT		StringRange
ca.cam1enable	CHOICE		EnumRange (DISABLE ENABLE)
ca.cam1type	TEXT	R/O	StringRange R/O
ca.cam2enable	CHOICE		EnumRange (DISABLE ENABLE)
ca.cam2type	TEXT	R/O	StringRange R/O
ca.casmode	CHOICE		EnumRange (AUTO MANUAL)
ca.casid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ca.casname	TEXT	R/O	StringRange R/O
ca.casmonitor	CHOICE		EnumRange (DISABLE ENABLE)
ipin.1.enable	CHOICE		EnumRange DIS (OFF ON)
ipin.1.address	TEXT		StringRange DIS

name	*type*	*flags*	*range*
ipin.1.port	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.1.interface	CHOICE		EnumRange DIS (1 2)
ipin.1.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.1.bufferDuration	INTEGER		IntegerRange DIS (1 .. 4000)
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.state	TEXT	R/O	StringRange R/O DIS
ipin.1.dataRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
ipin.1.buffer	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsReor	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsDupl	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.fecParity	TEXT	R/O	StringRange R/O DIS
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.enable	CHOICE		EnumRange DIS (OFF ON)
ipin.2.address	TEXT		StringRange DIS
ipin.2.port	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.2.interface	CHOICE		EnumRange DIS (1 2)
ipin.2.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.2.bufferDuration	INTEGER		IntegerRange DIS (1 .. 4095)
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.state	TEXT	R/O	StringRange R/O DIS
ipin.2.dataRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
ipin.2.buffer	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsReor	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
ipin.2.packetsDupl	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.fecParity	TEXT	R/O	StringRange R/O DIS
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Carrier lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Sync lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TS S2 present)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TS S2 error)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TS IP 1 Sync)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TS IP 1 Receive)
faults.09	ALARM	R/O	AlarmFlagRange R/O (TS IP 2 Sync)
faults.10	ALARM	R/O	AlarmFlagRange R/O (TS IP 2 Receive)
internal.band	CHOICE	R/O	EnumRange R/O (UNKNOWN LO HI)
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.asiId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ipId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.satId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ifc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.629 Sencore-MRD5800

This page describes the device driver and the device window for a Sencore IRD MRD 5800.

It supports up to 8 audio channels.

Note: As the unit does not allow to change LO or receive frequency out of the rf range in one SNMP command the usage of the device stored lo frequency can be switched OFF using config.useLoFreq. In this case the device stored LO Frequency will not be changed at any time by the driver and is used only for calculation. This mode is useful when operating the unit together with a switch matrix, a CDB and/or RX-Channel device.

It is tested with the following models:

- D9854_4x0 SW: 4.8.0
- D9800-SS-MPEGOIP SW: 4.8.0

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The Rf monitoring and control parameters.
- --- Contains IP in parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read

community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power
- lnbVoltage --- Selects the voltage level applied to the input if lnbPower is enabled.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter below becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- marginThreshold --- Defines the minimum s/n margin value the driver accepts as OK. s/n margin values below this threshold raise an "Margin" fault (faults.30). The margin fault detection is done by the sat-nms device driver, not by the device itself. Hence, the margin value ist stored as a saved parameter on the VLC rather than as a parameter in the IRD.
- audioChannels --- Defines the number of audio channels (2 or 4) of the unit. *useLoFreq --- As the unit does not allow to change lo or receive frequency out of the given rf range in one SNMP command the usage of the device stored lo frequency can be switched off using config.useLoFreq.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.psu	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.marginOffset	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.audioChannels	CHOICE	SAVE SETUP	EnumRange (4 8)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
internal.margin	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.interface	CHOICE	R/O	EnumRange R/O (IP ASI SAT)

name	*type*	*flags*	*range*
internal.input	CHOICE	R/O	EnumRange R/O (1 2 3 4)
ids2x1	TEXT	R/O	StringRange R/O
ids2x2	TEXT	R/O	StringRange R/O
idrf	TEXT	R/O	StringRange R/O
idDr	TEXT	R/O	StringRange R/O
idip1	TEXT	R/O	StringRange R/O
idip2	TEXT	R/O	StringRange R/O
idasi1	TEXT	R/O	StringRange R/O
idasi2	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
interface	CHOICE		EnumRange (IP ASI SAT)
input	CHOICE		EnumRange (1 2 3 4)
dvbmode	TEXT	R/O	StringRange R/O
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
symbolRateMode	CHOICE		EnumRange (AUTO MANUAL)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
polarization	CHOICE		EnumRange (X Y)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
serviceMode	CHOICE		EnumRange (AUTOMATIC SERVICE PIDLOCKED)

name	*type*	*flags*	*range*
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.margin	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.pid	INTEGER		IntegerRange (0 .. 0)
video.profile	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.colorType	TEXT	R/O	StringRange R/O
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitDepth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.1.arDownConvert	CHOICE		EnumRange (ANAMORPHIC PILLARBOX CENTER-CUT)
video.1.arUpConvert	CHOICE		EnumRange (ANAMORPHIC LETTERBOX CENTER-CUT)
video.2.arDownConvert	CHOICE		EnumRange (ANAMORPHIC PILLARBOX CENTER-CUT)
video.2.arUpConvert	CHOICE		EnumRange (ANAMORPHIC LETTERBOX CENTER-CUT)
video.1.failMode	CHOICE		EnumRange (FREEZE COLOR)
video.1.fail	CHOICE		EnumRange (BLACK WHITE YELLOW CYAN MAGENTA RED BLUE GREEN GRAY)

name	*type*	*flags*	*range*
video.2.failMode	CHOICE		EnumRange (FREEZE COLOR)
video.2.fail	CHOICE		EnumRange (BLACK WHITE YELLOW CYAN MAGENTA RED BLUE GREEN GRAY)
video.1.formatMode	CHOICE		EnumRange (AUTO MANUAL)
video.2.formatMode	CHOICE		EnumRange (AUTO MANUAL)
video.1.format	CHOICE		EnumRange (UNKNOWN SD-720x480i-4x3-29.97fps SD-720x480i-16x9-29.97fps SD-720x576i-4x3-25fps SD-720x576i-16x9-25fps HD-1280x720p-50fps HD-1280x720p-59.94fps HD-1280x720p-60fps HD-1920x1080i-25fps HD-1920x1080i-29.97fps HD-1920x1080i-30fps HD-1920x1080p-23.98fps HD-1920x1080p-24fps HD-1920x1080p-25fps HD-1920x1080p-29.97fps HD-1920x1080p-30fps)
video.2.format	CHOICE		EnumRange (UNKNOWN SD-720x480i-4x3-29.97fps SD-720x480i-16x9-29.97fps SD-720x576i-4x3-25fps SD-720x576i-16x9-25fps HD-1280x720p-50fps HD-1280x720p-59.94fps HD-1280x720p-60fps HD-1920x1080i-25fps HD-1920x1080i-29.97fps HD-1920x1080i-30fps HD-1920x1080p-23.98fps HD-1920x1080p-24fps HD-1920x1080p-25fps HD-1920x1080p-29.97fps HD-1920x1080p-30fps)
audio.1.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.1.delay	INTEGER		IntegerRange (-50 .. 100)
audio.1.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)

name	*type*	*flags*	*range*
audio.1.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.1.pid	TEXT		StringRange
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.2.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.2.delay	INTEGER		IntegerRange (-50 .. 100)
audio.2.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.2.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.2.pid	TEXT		StringRange
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.3.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.3.delay	INTEGER		IntegerRange (-50 .. 100)
audio.3.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.3.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.3.pid	TEXT		StringRange
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.4.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.4.delay	INTEGER		IntegerRange (-50 .. 100)
audio.4.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.4.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.4.pid	TEXT		StringRange
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.5.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.5.delay	INTEGER		IntegerRange (-50 .. 100)
audio.5.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.5.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.5.pid	TEXT		StringRange
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.6.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)

name	*type*	*flags*	*range*
audio.6.delay	INTEGER		IntegerRange (-50 .. 100)
audio.6.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.6.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.6.pid	TEXT		StringRange
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.7.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.7.delay	INTEGER		IntegerRange (-50 .. 100)
audio.7.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.7.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.7.pid	TEXT		StringRange
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.8.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.8.delay	INTEGER		IntegerRange (-50 .. 100)
audio.8.dolbyOffset	CHOICE		EnumRange (NONE 1-FRAME)
audio.8.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DUAL-MONO DUAL-LEFT DUAL-RIGHT DOLBY-SURR-Lt/Rt AUTO-Lt/Rt)
audio.8.pid	TEXT		StringRange

name	*type*	*flags*	*range*
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.coding	TEXT	R/O	StringRange R/O
sdi.1.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.2.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.3.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.4.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)

name	*type*	*flags*	*range*
sdi.5.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.6.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.7.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
sdi.8.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.1.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)

name	*type*	*flags*	*range*
aes.2.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.3.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.4.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.5.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.6.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)

name	*type*	*flags*	*range*
aes.7.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
aes.8.source	CHOICE		EnumRange (OFF 1-PASSTHROUGH 1-PCM 2-PASSTHROUGH 2-PCM 3-PASSTHROUGH 3-PCM 4-PASSTHROUGH 4-PCM 5-PASSTHROUGH 5-PCM 6-PASSTHROUGH 6-PCM 6-PASSTHROUGH 7-PCM 8-PASSTHROUGH 8-PCM)
ca.bissOperation	CHOICE		EnumRange (ALL SERVICE-ONLY DISABLED PID-ONLY)
ca.bissmode	CHOICE		EnumRange (BISS-1 BISS-E)
ca.bisskey	TEXT		StringRange
ipin.1.interface	CHOICE		EnumRange (PORT-1 PORT-2)
ipin.1.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 65535)
ipin.1.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.1.bufferDuration	INTEGER		IntegerRange (1 .. 4000)
ipin.1.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.1.type	CHOICE	R/O	EnumRange R/O (UDP RTP N/A)
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.reset	TEXT	NOPRESET	StringRange
ipin.2.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.interface	CHOICE		EnumRange (PORT-1 PORT-2)
ipin.2.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange
ipin.2.port	INTEGER		IntegerRange (0 .. 65535)
ipin.2.fecEnable	CHOICE		EnumRange (OFF ON)
ipin.2.bufferDuration	INTEGER		IntegerRange (1 .. 4000)
ipin.2.type	CHOICE	R/O	EnumRange R/O (UDP RTP N/A)
ipin.2.flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsReor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.buffer	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ipin.2.reset	TEXT	NOPRESET	StringRange
ipout.1.enable	CHOICE		EnumRange (OFF ON)
ipout.1.interface	CHOICE		EnumRange (PORT-1 PORT-2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.packets	INTEGER		IntegerRange (1 .. 7)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP)
ipout.2.enable	CHOICE		EnumRange (OFF ON)
ipout.2.interface	CHOICE		EnumRange (PORT-1 PORT-2)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 0)
ipout.2.packets	INTEGER		IntegerRange (0 .. 0)
ipout.2.protocol	CHOICE		EnumRange (UDP RTP)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.630 Sencore-MRD7000

This page describes the device driver and the device window for the Sencore MRD700 Multidecoder (IRD), controlled via SNMP.

The driver was tested with sw release**1.10.0**

The driver can only handle one decoding instance. As the MRD7000 is equipped with up to 4 decoders, each instance has to be setup in a single device. The respective instance has to be configured at first setup via config.decoder

The Sencore IRD only supports two service allocation methods.

- AUTO decodes the first program from the received service.
- SERVICE program has to be selected manually from the service list. It does not support any automatic service hunt mode when set to SERVICE.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general input and program selection.
- --- Contains RTP/UDP input parameters
- --- Contains SRT input parameters
- --- Contains ZIXI input parameters
- --- Contains the conditional access (BISS) parameters.
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the MRD7000.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
    - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
    - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
    - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters

recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- decoder --- Defines the desired decoding instance.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.decoder	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (OFF 13V 18V)

name	*type*	*flags*	*range*
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.hasIpOutput	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.deviceComment	TEXT	SAVE SETUP	StringRange
inputId	TEXT	R/O	StringRange R/O
inputId2	TEXT	R/O	StringRange R/O
tableId	TEXT	R/O	StringRange R/O
tableId2	TEXT	R/O	StringRange R/O
rfInputId	TEXT	R/O	StringRange R/O
int.h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
int.v	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.interface	CHOICE	R/O	EnumRange R/O (SAT IP FILE SRT SEAMLESS ZIXI N/A)
internal.input	CHOICE	R/O	EnumRange R/O (1 2)
internal.interface2	CHOICE	R/O	EnumRange R/O (IP FILE SRT SEAMLESS ZIXI N/A)
internal.input2	CHOICE	R/O	EnumRange R/O (1 2)
internal.biss	TEXT	R/O	StringRange R/O
internal.bissOid	TEXT	R/O	StringRange R/O
internal.programAuto	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.running	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.band	TEXT	R/O	StringRange R/O
internal.lo	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
interface	CHOICE		EnumRange (SAT IP SRT ZIXI)
interface.2	CHOICE		EnumRange (IP SRT ZIXI)
input	CHOICE		EnumRange (1 2 3 4)
input.2	CHOICE		EnumRange (1 2 3 4)
input.active	CHOICE	R/O	EnumRange R/O (MAIN BACKUP)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
dataRate2	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
input.1.interface	CHOICE		EnumRange (ETH0 ETH1)
input.2.interface	CHOICE		EnumRange (ETH0 ETH1)
input.1.enable	CHOICE		EnumRange (ON OFF)
input.2.enable	CHOICE		EnumRange (ON OFF)
input.redundancyMode	CHOICE		EnumRange (MANUAL TS-SYNC DECODE)
input.1.redundancyMode	CHOICE		EnumRange (MANUAL TS-SYNC DECODE)
input.2.redundancyMode	CHOICE		EnumRange (MANUAL TS-SYNC DECODE TS-RESTORE)
input.switchover	FLOAT		DoubleRange (1.0 .. 20.0)
input.manualSw	TEXT	NOPRESET	StringRange
serviceMode	CHOICE		EnumRange (AUTOMATIC SERVICE)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)

name	*type*	*flags*	*range*
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts2	BOOLEAN	R/O	BooleanRange R/O (true false)
ca.bissmode	CHOICE		EnumRange (BISS-1 BISS-E BISS-CA)
ca.bissmode2	CHOICE		EnumRange (BURIED INJECTED)
ca.bisskey	TEXT		StringRange
dvbmode	CHOICE		EnumRange (DVBS DVBS2)
modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK 64APSK 128APSK 256APSK UNKOWNN)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
pls	INTEGER		IntegerRange (0 .. 262141)
multistream	CHOICE		EnumRange (OFF ON)
isi	INTEGER		IntegerRange (0 .. 255)
state.modulation	TEXT	R/O	StringRange R/O
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.dvbmode	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pilots	TEXT	R/O	StringRange R/O
state.spectrum	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
polarization	CHOICE		EnumRange (X Y)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.snr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.profile	TEXT	R/O	StringRange R/O
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.colorType	TEXT	R/O	StringRange R/O
video.aspectRatio	TEXT	R/O	StringRange R/O
video.width	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.height	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitDepth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.failMode	CHOICE		EnumRange (FREEZE COLOR)
video.fail	CHOICE		EnumRange (BLACK WHITE YELLOW CYAN MAGENTA RED BLUE GREEN GRAY)
video.formatMode	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
video.format	CHOICE		EnumRange (UNKNOWN HD-1280x720p-50fps HD-1280x720p-59.94fps HD-1280x720p-60fps HD-1920x1080i-25fps HD-1920x1080i-29.97fps HD-1920x1080i-30fps HD-1920x1080p-23.98fps HD-1920x1080p-24fps HD-1920x1080p-25fps HD-1920x1080p-29.97fps HD-1920x1080p-30fps HD-1920x1080p-50fps HD-1920x1080p-59.94fps HD-1920x1080p-60fps 4K-3840x2160p-23.98fps 4K-3840x2160p-24fps 4K-3840x2160p-25fps 4K-3840x2160p-29.97fps 4K-3840x2160p-30fps 4K-3840x2160p-50fps 4K-3840x2160p-59.94fps 4K-3840x2160p-60fps)
audio.1.pid	TEXT	R/O	StringRange R/O
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.1.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.1.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.2.pid	TEXT	R/O	StringRange R/O
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.2.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.2.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.3.pid	TEXT	R/O	StringRange R/O
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.3.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.3.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.4.pid	TEXT	R/O	StringRange R/O
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.4.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)

name	*type*	*flags*	*range*
audio.4.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.5.pid	TEXT	R/O	StringRange R/O
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.5.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.5.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.6.pid	TEXT	R/O	StringRange R/O
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.6.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.6.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.7.pid	TEXT	R/O	StringRange R/O
audio.7.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.7.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.7.coding	TEXT	R/O	StringRange R/O
audio.7.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.7.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.7.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.8.pid	TEXT	R/O	StringRange R/O
audio.8.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.8.coding	TEXT	R/O	StringRange R/O
audio.8.dynamic	CHOICE		EnumRange (LINE RF CUSTOM-0 CUSTOM-1)
audio.8.processing	CHOICE		EnumRange (DOWNMIX DISCRETE)
audio.8.downmix	CHOICE		EnumRange (STEREO-Lo/Ro DOLBY-SURR-Lt/Rt AUTO-Lt/Rt DUAL-MONO DUAL-LEFT DUAL-RIGHT HEADPHONE SPEAKER)
audio.1.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.2.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.3.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.4.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.5.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)

name	*type*	*flags*	*range*
audio.6.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.7.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
audio.8.mode	CHOICE		EnumRange (PASSTHROUGH PCM AUTO)
sdi.1.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.2.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.3.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.4.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.5.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.6.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.7.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
sdi.8.source	CHOICE		EnumRange (OFF 1 2 3 4 5 6 7 8)
ipin.1.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange DIS
ipin.1.port	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.1.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.1.type	CHOICE	R/O	EnumRange R/O DIS (UDP RTP UNKNOWN)
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.packetsDupl	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
ipin.2.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange DIS
ipin.2.port	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.2.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.2.bufferDuration	INTEGER		IntegerRange DIS (1 .. 4000)
ipin.2.type	CHOICE	R/O	EnumRange R/O DIS (UDP RTP UNKNOWN)
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.packetsDupl	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.mode	CHOICE		EnumRange DIS (CALLER LISTENER RENDEZVOUS)
srt.1.address	TEXT		StringRange DIS
srt.1.port	INTEGER		IntegerRange DIS (0 .. 65535)
srt.1.localPortMode	CHOICE		EnumRange DIS (AUTO MANUAL)
srt.1.localPort	INTEGER		IntegerRange DIS (1024 .. 65535)
srt.1.timeout	FLOAT		DoubleRange DIS (0.0 .. 100.0)
srt.1.latency	INTEGER		IntegerRange DIS (20 .. 8000)
srt.1.key	TEXT		StringRange DIS
srt.1.state	TEXT	R/O	StringRange R/O DIS
srt.1.meas.localPort	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.encryption	TEXT	R/O	StringRange R/O DIS
srt.1.decryption	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
srt.1.rtt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.buffer	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.linkBw	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
srt.1.meas.latency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.reconnect	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.1.packetsSkip	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.mode	CHOICE		EnumRange DIS (CALLER LISTENER RENDEZVOUS)
srt.2.address	TEXT		StringRange DIS
srt.2.port	INTEGER		IntegerRange DIS (0 .. 65535)
srt.2.localPortMode	CHOICE		EnumRange DIS (AUTO MANUAL)
srt.2.localPort	INTEGER		IntegerRange DIS (1024 .. 65535)
srt.2.timeout	FLOAT		DoubleRange DIS (0.0 .. 100.0)
srt.2.latency	INTEGER		IntegerRange DIS (20 .. 8000)
srt.2.key	TEXT		StringRange DIS
srt.2.state	TEXT	R/O	StringRange R/O DIS
srt.2.meas.localPort	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.encryption	TEXT	R/O	StringRange R/O DIS
srt.2.decryption	TEXT	R/O	StringRange R/O DIS
srt.2.rtt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.buffer	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.linkBw	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
srt.2.meas.latency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
srt.2.reconnect	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
srt.2.packetsSkip	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.stream	TEXT		StringRange DIS
zixi.1.remote	TEXT		StringRange DIS
zixi.1.host	TEXT		StringRange DIS
zixi.1.port	INTEGER		IntegerRange DIS (1025 .. 65525)
zixi.1.password	TEXT		StringRange DIS
zixi.1.tls	CHOICE		EnumRange DIS (IGNORE NOT IGNORE)
zixi.1.latency	INTEGER		IntegerRange DIS (30 .. 10000)
zixi.1.decryption	CHOICE		EnumRange DIS (DISABLED AES128 AES192 AES256 AUTO)
zixi.1.key	TEXT		StringRange DIS
zixi.1.overhead	INTEGER		IntegerRange DIS (0 .. 50)
zixi.1.state	TEXT	R/O	StringRange R/O DIS
zixi.1.reconnect	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.rtt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.jitter	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsDrop	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsCorrFec	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsCorrArq	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.1.packetsDuplArq	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.stream	TEXT		StringRange DIS

name	*type*	*flags*	*range*
zixi.2.remote	TEXT		StringRange DIS
zixi.2.host	TEXT		StringRange DIS
zixi.2.port	INTEGER		IntegerRange DIS (1025 .. 65525)
zixi.2.password	TEXT		StringRange DIS
zixi.2.tls	CHOICE		EnumRange DIS (IGNORE NOT IGNORE)
zixi.2.latency	INTEGER		IntegerRange DIS (30 .. 10000)
zixi.2.decryption	CHOICE		EnumRange DIS (DISABLED AES128 AES192 AES256 AUTO)
zixi.2.key	TEXT		StringRange DIS
zixi.2.overhead	INTEGER		IntegerRange DIS (0 .. 50)
zixi.2.state	TEXT	R/O	StringRange R/O DIS
zixi.2.reconnect	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.rtt	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.jitter	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsDrop	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsCorrFec	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsCorrArq	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
zixi.2.packetsDuplArq	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
reset	TEXT	NOPRESET	StringRange
ipin.1.reset	TEXT	NOPRESET	StringRange
ipin.2.reset	TEXT	NOPRESET	StringRange
srt.1.reset	TEXT	NOPRESET	StringRange
srt.2.reset	TEXT	NOPRESET	StringRange
zixi.1.reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
zixi.2.reset	TEXT	NOPRESET	StringRange
ipout.1.enable	CHOICE		EnumRange DIS (OFF ON)
ipout.1.descramble	CHOICE		EnumRange DIS (OFF ON)
ipout.1.interface	CHOICE		EnumRange DIS (ETH0 ETH1)
ipout.1.address	TEXT		StringRange DIS
ipout.1.port	INTEGER		IntegerRange DIS (1025 .. 65535)
ipout.1.sourceMode	CHOICE		EnumRange DIS (AUTO MANUAL)
ipout.1.sourceAddress	TEXT		StringRange DIS
ipout.1.sourcePort	INTEGER		IntegerRange DIS (1025 .. 65535)
ipout.1.packets	INTEGER		IntegerRange DIS (1 .. 7)
ipout.1.protocol	CHOICE		EnumRange DIS (RTP UDP)
ipout.1.fecEnable	CHOICE		EnumRange DIS (OFF 1D 2D)
ipout.1.fecColumns	INTEGER		IntegerRange DIS (1 .. 20)
ipout.1.fecRows	INTEGER		IntegerRange DIS (4 .. 20)
ipout.1.datarate	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ipout.1.state	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.631 ServerTech-PDU-Sentry-3

This page describes the device driver and the device window for the for the ServerTech Switched PDU Sentry 3 using SNMP.

The driver is prepared to control up to 24 outlets.

It was tested with:

- model: Sentry Switched CDU POPS PIPS -> Sentry-3
- firmware version: Sentry Switched CDU Version 7.1c

For newer models or firmware you might use:

- ServerTech-PDU-Sentry-4.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Outlets overview and settings.
- --- Lines page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- sequenceInterval --- Defines the power-on sequencing interval for all outlets in seconds.
- rebootDelay --- Defines the delay in seconds for all outlets.
- temp.loThreshold.xx --- Defines the low temperature alarm for the individual sensors.

- temp.hiThreshold.xx --- Defines the high temperature alarm for the individual sensors.
- humidity.loThreshold.xx --- Defines the low humidity alarm for the individual sensors.
- humidity.hiThreshold.xx --- Defines the high humidity alarm for the individual sensors.
- sensUnit --- Switch between temperature display Fahrenheit and Celsius.
- postOnDelay.xx --- Defines the post on switch delay individual for each outlet.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.model	TEXT	R/O	StringRange R/O
info.outlets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.sequenceInterval	INTEGER	SETUP	IntegerRange (1 .. 15)
config.rebootDelay	INTEGER	SETUP	IntegerRange (5 .. 600)
config.loadHiTreshold	INTEGER	SETUP	IntegerRange (5 .. 600)
config.temp.loThreshold.01	INTEGER	SETUP	IntegerRange (0 .. 254)
config.temp.loThreshold.02	INTEGER	SETUP	IntegerRange (0 .. 254)

name	*type*	*flags*	*range*
config.temp.hiThreshold.01	INTEGER	SETUP	IntegerRange (0 .. 254)
config.temp.hiThreshold.02	INTEGER	SETUP	IntegerRange (0 .. 254)
config.humidity.loThreshold.01	INTEGER	SETUP	IntegerRange (0 .. 100)
config.humidity.loThreshold.02	INTEGER	SETUP	IntegerRange (0 .. 100)
config.humidity.hiThreshold.01	INTEGER	SETUP	IntegerRange (0 .. 100)
config.humidity.hiThreshold.02	INTEGER	SETUP	IntegerRange (0 .. 100)
config.sensUnit	CHOICE	SETUP	EnumRange (CELSIUS FAHRENHEIT)
config.postOnDelay.01	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.02	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.03	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.04	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.05	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.06	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.07	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.08	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.09	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.10	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.11	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.12	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.13	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.14	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.15	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.16	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.17	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.18	INTEGER	SETUP	IntegerRange (0 .. 900)

name	*type*	*flags*	*range*
config.postOnDelay.19	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.20	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.21	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.22	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.23	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.24	INTEGER	SETUP	IntegerRange (0 .. 900)
internal.xx.powerFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.xx.crestFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.sensorStatus.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sensorStatus.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.temperature.01.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
internal.temperature.02.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
internal.humidity.01.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
internal.humidity.02.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
input.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.state	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)

name	*type*	*flags*	*range*
input.loadState	CHOICE	R/O	EnumRange R/O (NORMAL NOT ON READING LOAD LOW LOAD HIGH OVER LOAD READ ERROR NO COMM)
input.load	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.capacityLoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.power	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.appPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.powerFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
input.crestFactor	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.energy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.reactance	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
input.capacityLoadUsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.capacityVA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.capacityVAUsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
name.01	TEXT		StringRange
name.02	TEXT		StringRange
name.03	TEXT		StringRange
name.04	TEXT		StringRange
name.05	TEXT		StringRange
name.06	TEXT		StringRange
name.07	TEXT		StringRange
name.08	TEXT		StringRange
name.09	TEXT		StringRange
name.10	TEXT		StringRange
name.11	TEXT		StringRange

name	*type*	*flags*	*range*
name.12	TEXT		StringRange
name.13	TEXT		StringRange
name.14	TEXT		StringRange
name.15	TEXT		StringRange
name.16	TEXT		StringRange
name.17	TEXT		StringRange
name.18	TEXT		StringRange
name.19	TEXT		StringRange
name.20	TEXT		StringRange
name.21	TEXT		StringRange
name.22	TEXT		StringRange
name.23	TEXT		StringRange
name.24	TEXT		StringRange
status.01	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.02	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.03	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.04	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.05	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)

name	*type*	*flags*	*range*
status.06	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.07	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.08	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.09	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.10	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.11	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.12	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.13	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.14	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)

name	*type*	*flags*	*range*
status.15	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.16	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.17	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.18	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.19	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.20	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.21	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.22	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
status.23	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)

name	*type*	*flags*	*range*
status.24	CHOICE	R/O	EnumRange R/O (OFF ON OFF WAIT ON WAIT OFF ERROR ON ERROR NO COMM READING OFF FUSE ON FUSE)
load.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
load.01.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.02.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.03.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.04.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.05.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.06.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.07.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.08.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.09.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.10.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.11.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.12.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.13.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.14.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.15.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.16.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.17.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.18.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.19.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.20.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.21.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.22.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.23.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.24.lowLimit	INTEGER		IntegerRange (0 .. 255)
load.01.highLimit	INTEGER		IntegerRange (0 .. 255)
load.02.highLimit	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
load.03.highLimit	INTEGER		IntegerRange (0 .. 255)
load.04.highLimit	INTEGER		IntegerRange (0 .. 255)
load.05.highLimit	INTEGER		IntegerRange (0 .. 255)
load.06.highLimit	INTEGER		IntegerRange (0 .. 255)
load.07.highLimit	INTEGER		IntegerRange (0 .. 255)
load.08.highLimit	INTEGER		IntegerRange (0 .. 255)
load.09.highLimit	INTEGER		IntegerRange (0 .. 255)
load.10.highLimit	INTEGER		IntegerRange (0 .. 255)
load.11.highLimit	INTEGER		IntegerRange (0 .. 255)
load.12.highLimit	INTEGER		IntegerRange (0 .. 255)
load.13.highLimit	INTEGER		IntegerRange (0 .. 255)
load.14.highLimit	INTEGER		IntegerRange (0 .. 255)
load.15.highLimit	INTEGER		IntegerRange (0 .. 255)
load.16.highLimit	INTEGER		IntegerRange (0 .. 255)
load.17.highLimit	INTEGER		IntegerRange (0 .. 255)
load.18.highLimit	INTEGER		IntegerRange (0 .. 255)
load.19.highLimit	INTEGER		IntegerRange (0 .. 255)
load.20.highLimit	INTEGER		IntegerRange (0 .. 255)
load.21.highLimit	INTEGER		IntegerRange (0 .. 255)
load.22.highLimit	INTEGER		IntegerRange (0 .. 255)
load.23.highLimit	INTEGER		IntegerRange (0 .. 255)
load.24.highLimit	INTEGER		IntegerRange (0 .. 255)

name	*type*	*flags*	*range*
controlState.01	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.02	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.03	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.04	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.05	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.06	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.07	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.08	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.09	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.10	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.11	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.12	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.13	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.14	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.15	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.16	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.17	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.18	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.19	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.20	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.21	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.22	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.23	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.24	CHOICE	R/O	EnumRange R/O (IDLE OFF IDLE ON WAKE OFF WAKE ON OFF ON LOCKED OFF LOCKED OON REBOOT SHUTDOWN PEND ON PEND OFF MINIMUM OFF MINIMUM ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
pulse.01	CHOICE		EnumRange (REBOOT)
pulse.02	CHOICE		EnumRange (REBOOT)
pulse.03	CHOICE		EnumRange (REBOOT)
pulse.04	CHOICE		EnumRange (REBOOT)
pulse.05	CHOICE		EnumRange (REBOOT)
pulse.06	CHOICE		EnumRange (REBOOT)

name	*type*	*flags*	*range*
pulse.07	CHOICE		EnumRange (REBOOT)
pulse.08	CHOICE		EnumRange (REBOOT)
pulse.09	CHOICE		EnumRange (REBOOT)
pulse.10	CHOICE		EnumRange (REBOOT)
pulse.11	CHOICE		EnumRange (REBOOT)
pulse.12	CHOICE		EnumRange (REBOOT)
pulse.13	CHOICE		EnumRange (REBOOT)
pulse.14	CHOICE		EnumRange (REBOOT)
pulse.15	CHOICE		EnumRange (REBOOT)
pulse.16	CHOICE		EnumRange (REBOOT)
pulse.17	CHOICE		EnumRange (REBOOT)
pulse.18	CHOICE		EnumRange (REBOOT)
pulse.19	CHOICE		EnumRange (REBOOT)
pulse.20	CHOICE		EnumRange (REBOOT)
pulse.21	CHOICE		EnumRange (REBOOT)
pulse.22	CHOICE		EnumRange (REBOOT)
pulse.23	CHOICE		EnumRange (REBOOT)
pulse.24	CHOICE		EnumRange (REBOOT)
state.01	CHOICE		EnumRange (NONE ON OFF)
state.02	CHOICE		EnumRange (NONE ON OFF)
state.03	CHOICE		EnumRange (NONE ON OFF)
state.04	CHOICE		EnumRange (NONE ON OFF)
state.05	CHOICE		EnumRange (NONE ON OFF)
state.06	CHOICE		EnumRange (NONE ON OFF)
state.07	CHOICE		EnumRange (NONE ON OFF)
state.08	CHOICE		EnumRange (NONE ON OFF)
state.09	CHOICE		EnumRange (NONE ON OFF)

name	*type*	*flags*	*range*
state.10	CHOICE		EnumRange (NONE ON OFF)
state.11	CHOICE		EnumRange (NONE ON OFF)
state.12	CHOICE		EnumRange (NONE ON OFF)
state.13	CHOICE		EnumRange (NONE ON OFF)
state.14	CHOICE		EnumRange (NONE ON OFF)
state.15	CHOICE		EnumRange (NONE ON OFF)
state.16	CHOICE		EnumRange (NONE ON OFF)
state.17	CHOICE		EnumRange (NONE ON OFF)
state.18	CHOICE		EnumRange (NONE ON OFF)
state.19	CHOICE		EnumRange (NONE ON OFF)
state.20	CHOICE		EnumRange (NONE ON OFF)
state.21	CHOICE		EnumRange (NONE ON OFF)
state.22	CHOICE		EnumRange (NONE ON OFF)
state.23	CHOICE		EnumRange (NONE ON OFF)
state.24	CHOICE		EnumRange (NONE ON OFF)
capacity.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
capacity.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
voltage.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
power.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
appPower.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
powerFactor.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
powerFactor.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
crestFactor.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
energy.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
wakeUpState.01	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.02	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.03	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.04	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.05	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.06	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.07	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.08	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.09	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.10	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.11	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.12	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.13	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.14	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.15	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.16	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.17	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.18	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.19	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.20	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.21	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.22	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.23	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.24	CHOICE		EnumRange (LAST OFF ON)
name.sens.01	TEXT		StringRange

name	*type*	*flags*	*range*
name.sens.02	TEXT		StringRange
temperature.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.01.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
temperature.02.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
humidity.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity.01.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
humidity.02.state	CHOICE	R/O	EnumRange R/O (NORMAL NOT FOUND READING TEMP LOW TEMP HIGH READ ERROR LOST NO COMM)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Input High Load)
faults.91	ALARM	R/O	AlarmFlagRange R/O (Input Low Load)
faults.98	ALARM	R/O	AlarmFlagRange R/O (Input Over Load)
faults.93	ALARM	R/O	AlarmFlagRange R/O (Temp Sensor 1)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Humidity Sensor 1)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Temp Sensor 2)

name	*type*	*flags*	*range*
faults.97	ALARM	R/O	AlarmFlagRange R/O (Humidity Sensor 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 Low Load)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 High Load)
faults.03	ALARM	R/O	AlarmFlagRange R/O (0utlet 1 Over Load)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 Low Load)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 High Load)
faults.06	ALARM	R/O	AlarmFlagRange R/O (0utlet 2 Over Load)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Outlet 3 Low Load)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Outlet 3 High Load)
faults.09	ALARM	R/O	AlarmFlagRange R/O (0utlet 3 Over Load)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 Low Load)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 High Load)
faults.12	ALARM	R/O	AlarmFlagRange R/O (0utlet 4 Over Load)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 Low Load)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 High Load)
faults.15	ALARM	R/O	AlarmFlagRange R/O (0utlet 5 Over Load)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Outlet 6 Low Load)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Outlet 6 High Load)
faults.18	ALARM	R/O	AlarmFlagRange R/O (0utlet 6 Over Load)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 Low Load)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 High Load)
faults.21	ALARM	R/O	AlarmFlagRange R/O (0utlet 7 Over Load)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 Low Load)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 High Load)
faults.24	ALARM	R/O	AlarmFlagRange R/O (0utlet 8 Over Load)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Outlet 9 Low Load)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Outlet 9 High Load)
faults.27	ALARM	R/O	AlarmFlagRange R/O (0utlet 9 Over Load)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 Low Load)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 High Load)
faults.30	ALARM	R/O	AlarmFlagRange R/O (0utlet 10 Over Load)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 Low Load)

name	*type*	*flags*	*range*
faults.32	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 High Load)
faults.33	ALARM	R/O	AlarmFlagRange R/O (0utlet 11 Over Load)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Outlet 12 Low Load)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Outlet 12 High Load)
faults.36	ALARM	R/O	AlarmFlagRange R/O (0utlet 12 Over Load)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 Low Load)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 High Load)
faults.39	ALARM	R/O	AlarmFlagRange R/O (0utlet 13 Over Load)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 Low Load)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 High Load)
faults.42	ALARM	R/O	AlarmFlagRange R/O (0utlet 14 Over Load)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Outlet 15 Low Load)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Outlet 15 High Load)
faults.45	ALARM	R/O	AlarmFlagRange R/O (0utlet 15 Over Load)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 Low Load)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 High Load)

name	*type*	*flags*	*range*
faults.48	ALARM	R/O	AlarmFlagRange R/O (0utlet 16 Over Load)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 Low Load)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 High Load)
faults.51	ALARM	R/O	AlarmFlagRange R/O (0utlet 17 Over Load)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Outlet 18 Low Load)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Outlet 18 High Load)
faults.54	ALARM	R/O	AlarmFlagRange R/O (0utlet 18 Over Load)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 Low Load)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 High Load)
faults.57	ALARM	R/O	AlarmFlagRange R/O (0utlet 19 Over Load)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 Low Load)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 High Load)
faults.60	ALARM	R/O	AlarmFlagRange R/O (0utlet 20 Over Load)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Outlet 21 Low Load)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Outlet 21 High Load)
faults.63	ALARM	R/O	AlarmFlagRange R/O (0utlet 21 Over Load)

name	*type*	*flags*	*range*
faults.64	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 Low Load)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 High Load)
faults.66	ALARM	R/O	AlarmFlagRange R/O (0utlet 22 Over Load)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 Low Load)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 High Load)
faults.69	ALARM	R/O	AlarmFlagRange R/O (0utlet 23 Over Load)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Outlet 24 Low Load)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Outlet 24 High Load)
faults.72	ALARM	R/O	AlarmFlagRange R/O (0utlet 24 Over Load)

9.632 ServerTech-PDU-Sentry-4

This page describes the device driver and the device window for the for the ServerTech Switched PDU Sentry 4 using SNMP.

The driver is prepared to control up to 24 outlets.

It was tested with:

- model: Sentry Switched CDU POPS PIPS -> Sentry-4
- firmware version: Sentry Switched CDU Version 8.0

For older models or firmware you might use:

- ServerTech-PDU-Sentry-3.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Outlets overview.
- --- Outlet setup page.
- --- Lines page.
- --- Cords page.
- --- Phases page.
- --- Branches page.
- --- Overcurrent Protection Page.
- --- The sensor measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter* --- *description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- sequenceInterval --- Defines the power-on sequencing interval for all outlets in seconds.
- rebootDelay --- Defines the delay in seconds for all outlets.
- temp.loThreshold.xx --- Defines the low temperature alarm for the individual sensors.
- temp.hiThreshold.xx --- Defines the high temperature alarm for the individual sensors.
- sensUnit --- Switch between temperature display Fahrenheit and Celsius.
- postOnDelay.xx --- Defines the post on switch delay individual for each outlet.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n"

character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.model	TEXT	R/O	StringRange R/O
info.outlets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.location	TEXT	R/O	StringRange R/O
config.sequenceInterval	INTEGER	SETUP	IntegerRange (1 .. 15)
config.rebootDelay	INTEGER	SETUP	IntegerRange (5 .. 600)
config.temp.loThreshold.01	INTEGER	R/O	IntegerRange R/O (-40 .. 253)
config.temp.loThreshold.02	INTEGER	R/O	IntegerRange R/O (-40 .. 253)
config.temp.hiThreshold.01	INTEGER	R/O	IntegerRange R/O (-40 .. 253)
config.temp.hiThreshold.02	INTEGER	R/O	IntegerRange R/O (-40 .. 253)
config.sensUnit	CHOICE	SETUP	EnumRange (CELSIUS FAHRENHEIT)
config.postOnDelay.01	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.02	INTEGER	SETUP	IntegerRange (0 .. 900)

name	*type*	*flags*	*range*
config.postOnDelay.03	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.04	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.05	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.06	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.07	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.08	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.09	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.10	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.11	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.12	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.13	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.14	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.15	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.16	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.17	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.18	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.19	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.20	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.21	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.22	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.23	INTEGER	SETUP	IntegerRange (0 .. 900)
config.postOnDelay.24	INTEGER	SETUP	IntegerRange (0 .. 900)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.xx.powerFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.xx.crestFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.sensorStatus.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.sensorStatus.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.status.01	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.02	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.03	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.04	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.11	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.12	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.13	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.status.14	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.humidity.01.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
internal.humidity.02.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
internal.temperature.01.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
internal.temperature.02.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
cord.nominalVoltage	INTEGER		IntegerRange (0 .. 600)
cord.currentCapacity	INTEGER		IntegerRange (0 .. 600)
cord.id	TEXT	R/O	StringRange R/O
cord.label	TEXT	R/O	StringRange R/O
cord.inletType	TEXT	R/O	StringRange R/O
cord.powerCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cord.state	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
cord.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
cord.activePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cord.activePowerStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
cord.appPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cord.appPowerStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
cord.powerUsed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
cord.powerFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
cord.powerFactorStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
cord.energy	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
cord.frequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cord.activePwrLowWarning	INTEGER		IntegerRange (0 .. 50000)
cord.activePwrHighWarning	INTEGER		IntegerRange (0 .. 50000)
cord.activePwrLowAlarm	INTEGER		IntegerRange (0 .. 50000)
cord.activePwrHighAlarm	INTEGER		IntegerRange (0 .. 50000)
cord.appPwrLowWarning	INTEGER		IntegerRange (0 .. 50000)
cord.appPwrHighWarning	INTEGER		IntegerRange (0 .. 50000)
cord.appPwrLowAlarm	INTEGER		IntegerRange (0 .. 50000)
cord.appPwrHighAlarm	INTEGER		IntegerRange (0 .. 50000)
cord.pwrFactorLowAlarm	FLOAT		DoubleRange (0.00 .. 1.00)
cord.pwrFactorLowWarning	FLOAT		DoubleRange (0.00 .. 1.00)

name	*type*	*flags*	*range*
phase.id	TEXT	R/O	StringRange R/O
phase.label	TEXT	R/O	StringRange R/O
phase.nominalVoltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
phase.state	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
phase.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
phase.voltage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
phase.voltageStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
phase.deviation	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
phase.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
phase.crestFactor	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
phase.activePower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
phase.appPower	INTEGER	R/O	IntegerRange R/O (0 .. 0)
phase.powerFactor	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
phase.powerFactorStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
phase.reactance	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
phase.energy	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
phase.voltageLowWarning	FLOAT		DoubleRange (0.0 .. 600.0)
phase.voltageHighWarning	FLOAT		DoubleRange (0.0 .. 600.0)
phase.voltageLowAlarm	FLOAT		DoubleRange (0.0 .. 600.0)
phase.voltageHighAlarm	FLOAT		DoubleRange (0.0 .. 600.0)
phase.pwrFactorLowAlarm	FLOAT		DoubleRange (0.00 .. 1.00)
phase.pwrFactorLowWarning	FLOAT		DoubleRange (0.00 .. 1.00)
name.01	TEXT		StringRange
name.02	TEXT		StringRange
name.03	TEXT		StringRange
name.04	TEXT		StringRange
name.05	TEXT		StringRange
name.06	TEXT		StringRange
name.07	TEXT		StringRange
name.08	TEXT		StringRange
name.09	TEXT		StringRange
name.10	TEXT		StringRange
name.11	TEXT		StringRange

name	*type*	*flags*	*range*
name.12	TEXT		StringRange
name.13	TEXT		StringRange
name.14	TEXT		StringRange
name.15	TEXT		StringRange
name.16	TEXT		StringRange
name.17	TEXT		StringRange
name.18	TEXT		StringRange
name.19	TEXT		StringRange
name.20	TEXT		StringRange
name.21	TEXT		StringRange
name.22	TEXT		StringRange
name.23	TEXT		StringRange
name.24	TEXT		StringRange
status.01	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.02	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.03	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.04	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.05	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.06	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.07	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.08	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)

name	*type*	*flags*	*range*
status.09	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.10	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.11	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.12	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.13	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.14	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.15	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.16	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.17	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.18	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.19	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.20	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.21	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.22	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.23	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
status.24	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
load.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
load.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
load.01.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.02.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.03.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.04.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)

name	*type*	*flags*	*range*
load.05.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.06.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.07.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.08.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.09.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.10.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.11.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.12.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.13.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.14.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.15.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.16.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.17.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.18.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.19.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.20.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.21.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.22.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.23.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.24.lowLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.01.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.02.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.03.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.04.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.05.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)

name	*type*	*flags*	*range*
load.06.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.07.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.08.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.09.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.10.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.11.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.12.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.13.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.14.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.15.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.16.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.17.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.18.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.19.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.20.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.21.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.22.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.23.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.24.highLimit	FLOAT		DoubleRange (0.0 .. 125.0)
load.01.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.02.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.03.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.04.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.05.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.06.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.07.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)

name	*type*	*flags*	*range*
load.08.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.09.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.10.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.11.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.12.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.13.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.14.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.15.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.16.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.17.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.18.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.19.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.20.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.21.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.22.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.23.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.24.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.01.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.02.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.03.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.04.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.05.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.06.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.07.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.08.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.09.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)

name	*type*	*flags*	*range*
load.10.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.11.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.12.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.13.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.14.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.15.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.16.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.17.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.18.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.19.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.20.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.21.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.22.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.23.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
load.24.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
loadStatus.01	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.02	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.03	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.04	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.05	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.06	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.07	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.08	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.09	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.10	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.11	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.12	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.13	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.14	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.15	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.16	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.17	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.18	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.19	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.20	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.21	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.22	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
loadStatus.23	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
loadStatus.24	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.01	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.02	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.03	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.04	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.05	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.06	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.07	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.08	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.09	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.10	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.11	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.12	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.13	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.14	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.15	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.16	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.17	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.18	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.19	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.20	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.21	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.22	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
activePowerStatus.23	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
activePowerStatus.24	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
controlState.01	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.02	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.03	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.04	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.05	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.06	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.07	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.08	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.09	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.10	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.11	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.12	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.13	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.14	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.15	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.16	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.17	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.18	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.19	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.20	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.21	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.22	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
controlState.23	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)

name	*type*	*flags*	*range*
controlState.24	CHOICE	R/O	EnumRange R/O (NOT SET FIXED ON IDLE OFF IDLE ON WAKE OFF WAKE ON OCP OFF OCP ON PEND ON PEND OFF OFF ON REBOO SHUTDOWN LOCKED OFF LOCKED ON EVENT OFF EVENT ON EVENT REBOOT EVENT SHUTDOWN)
outletStatus.01	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.02	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.03	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.04	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.05	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.06	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.07	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.08	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.09	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.10	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.11	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.12	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.13	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.14	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.15	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.16	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.17	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.18	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.19	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.20	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.21	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.22	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
outletStatus.23	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
outletStatus.24	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.01	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.02	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.03	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.04	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.05	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.06	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.07	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.08	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.09	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.10	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.11	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.12	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.13	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.14	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.15	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.16	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.17	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.18	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.19	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.20	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.21	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.22	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
powerFactorStatus.23	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
powerFactorStatus.24	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
pulse.01	CHOICE		EnumRange (REBOOT)
pulse.02	CHOICE		EnumRange (REBOOT)
pulse.03	CHOICE		EnumRange (REBOOT)
pulse.04	CHOICE		EnumRange (REBOOT)
pulse.05	CHOICE		EnumRange (REBOOT)
pulse.06	CHOICE		EnumRange (REBOOT)
pulse.07	CHOICE		EnumRange (REBOOT)
pulse.08	CHOICE		EnumRange (REBOOT)
pulse.09	CHOICE		EnumRange (REBOOT)
pulse.10	CHOICE		EnumRange (REBOOT)
pulse.11	CHOICE		EnumRange (REBOOT)
pulse.12	CHOICE		EnumRange (REBOOT)
pulse.13	CHOICE		EnumRange (REBOOT)
pulse.14	CHOICE		EnumRange (REBOOT)
pulse.15	CHOICE		EnumRange (REBOOT)
pulse.16	CHOICE		EnumRange (REBOOT)
pulse.17	CHOICE		EnumRange (REBOOT)
pulse.18	CHOICE		EnumRange (REBOOT)
pulse.19	CHOICE		EnumRange (REBOOT)
pulse.20	CHOICE		EnumRange (REBOOT)

name	*type*	*flags*	*range*
pulse.21	CHOICE		EnumRange (REBOOT)
pulse.22	CHOICE		EnumRange (REBOOT)
pulse.23	CHOICE		EnumRange (REBOOT)
pulse.24	CHOICE		EnumRange (REBOOT)
state.01	CHOICE		EnumRange (NONE ON OFF)
state.02	CHOICE		EnumRange (NONE ON OFF)
state.03	CHOICE		EnumRange (NONE ON OFF)
state.04	CHOICE		EnumRange (NONE ON OFF)
state.05	CHOICE		EnumRange (NONE ON OFF)
state.06	CHOICE		EnumRange (NONE ON OFF)
state.07	CHOICE		EnumRange (NONE ON OFF)
state.08	CHOICE		EnumRange (NONE ON OFF)
state.09	CHOICE		EnumRange (NONE ON OFF)
state.10	CHOICE		EnumRange (NONE ON OFF)
state.11	CHOICE		EnumRange (NONE ON OFF)
state.12	CHOICE		EnumRange (NONE ON OFF)
state.13	CHOICE		EnumRange (NONE ON OFF)
state.14	CHOICE		EnumRange (NONE ON OFF)
state.15	CHOICE		EnumRange (NONE ON OFF)
state.16	CHOICE		EnumRange (NONE ON OFF)
state.17	CHOICE		EnumRange (NONE ON OFF)
state.18	CHOICE		EnumRange (NONE ON OFF)
state.19	CHOICE		EnumRange (NONE ON OFF)
state.20	CHOICE		EnumRange (NONE ON OFF)
state.21	CHOICE		EnumRange (NONE ON OFF)
state.22	CHOICE		EnumRange (NONE ON OFF)

name	*type*	*flags*	*range*
state.23	CHOICE		EnumRange (NONE ON OFF)
state.24	CHOICE		EnumRange (NONE ON OFF)
capacity.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
capacity.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
voltage.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
power.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
power.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.03	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.04	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.05	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
appPower.06	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.07	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.09	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.13	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.14	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.15	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.16	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.17	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.18	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.19	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.20	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.21	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.22	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.23	INTEGER	R/O	IntegerRange R/O (0 .. 0)
appPower.24	INTEGER	R/O	IntegerRange R/O (0 .. 0)
powerFactor.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
powerFactor.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
powerFactor.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
crestFactor.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
crestFactor.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.01	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.02	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.03	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.04	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.05	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.06	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.07	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.08	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.09	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.10	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.11	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
energy.12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.13	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.14	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.15	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.16	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.17	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.18	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.19	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.20	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.22	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.23	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
energy.24	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
wakeUpState.01	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.02	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.03	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.04	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.05	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.06	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.07	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.08	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.09	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.10	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.11	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.12	CHOICE		EnumRange (LAST OFF ON)

name	*type*	*flags*	*range*
wakeUpState.13	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.14	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.15	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.16	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.17	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.18	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.19	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.20	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.21	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.22	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.23	CHOICE		EnumRange (LAST OFF ON)
wakeUpState.24	CHOICE		EnumRange (LAST OFF ON)
reactance.01	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.02	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.03	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.04	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.05	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.06	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)

name	*type*	*flags*	*range*
reactance.07	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.08	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.09	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.10	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.11	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.12	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.13	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.14	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.15	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.16	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.17	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.18	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)

name	*type*	*flags*	*range*
reactance.19	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.20	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.21	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.22	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.23	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
reactance.24	CHOICE	R/O	EnumRange R/O (UNKNOWN CAPACITIVE INDUCTIVE RESISTIVE)
name.sens.01	TEXT		StringRange
name.sens.02	TEXT		StringRange
temperature.01	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.02	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.01.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)

name	*type*	*flags*	*range*
temperature.02.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
humidity.01	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity.02	INTEGER	R/O	IntegerRange R/O (0 .. 0)
humidity.01.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
humidity.02.state	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
branch.01.id	TEXT	R/O	StringRange R/O
branch.02.id	TEXT	R/O	StringRange R/O
branch.01.label	TEXT	R/O	StringRange R/O
branch.02.label	TEXT	R/O	StringRange R/O
branch.01.currentCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
branch.02.currentCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
branch.01.phaseId	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
branch.02.phaseId	TEXT	R/O	StringRange R/O
branch.01.ocpId	TEXT	R/O	StringRange R/O
branch.02.ocpId	TEXT	R/O	StringRange R/O
branch.01.outletCount	INTEGER	R/O	IntegerRange R/O (1 .. 128)
branch.02.outletCount	INTEGER	R/O	IntegerRange R/O (1 .. 128)
branch.01.state	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
branch.02.state	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
branch.01.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
branch.02.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
branch.01.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
branch.02.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
branch.01.currentStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
branch.02.currentStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
branch.01.currentUtilized	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
branch.02.currentUtilized	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
branch.hysteresis	FLOAT		DoubleRange (0.0 .. 100.0)
branch.01.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
branch.02.lowWarning	FLOAT		DoubleRange (0.0 .. 125.0)
branch.01.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
branch.02.highWarning	FLOAT		DoubleRange (0.0 .. 125.0)
branch.01.lowAlarm	FLOAT		DoubleRange (0.0 .. 125.0)
branch.02.lowAlarm	FLOAT		DoubleRange (0.0 .. 125.0)
branch.01.highAlarm	FLOAT		DoubleRange (0.0 .. 125.0)
branch.02.highAlarm	FLOAT		DoubleRange (0.0 .. 125.0)
ocp.01.id	TEXT	R/O	StringRange R/O
ocp.02.id	TEXT	R/O	StringRange R/O
ocp.01.label	TEXT	R/O	StringRange R/O
ocp.02.label	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
ocp.01.type	CHOICE	R/O	EnumRange R/O (FUSE BREAKER)
ocp.02.type	CHOICE	R/O	EnumRange R/O (FUSE BREAKER)
ocp.01.currentCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ocp.02.currentCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ocp.01.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
ocp.02.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
line.id	TEXT	R/O	StringRange R/O
line.label	TEXT	R/O	StringRange R/O
line.currentCapacity	INTEGER	R/O	IntegerRange R/O (0 .. 0)
line.state	CHOICE	R/O	EnumRange R/O (UNKNOWN ON OFF)
line.status	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
line.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
line.currentStatus	CHOICE	R/O	EnumRange R/O (NORMAL DISABLED PURGED READING SETTLE NOT FOUND LOST READ ERROR NO COMM PWR ERROR BREAKER TRIPPED FUSE BLOWN LOW ALARM LOW WARNING HIGH WARNING HIGH ALARM ALARM UNDER LIMIT OVER LIMIT NVM FAIL PROFILE ERROR CONFLICT)
line.currentUtilized	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
line.lowWarning	FLOAT		DoubleRange (0.0 .. 600.0)
line.highWarning	FLOAT		DoubleRange (0.0 .. 600.0)
line.lowAlarm	FLOAT		DoubleRange (0.0 .. 600.0)
line.highAlarm	FLOAT		DoubleRange (0.0 .. 600.0)
faults.310	ALARM	R/O	AlarmFlagRange R/O (Line)
faults.320	ALARM	R/O	AlarmFlagRange R/O (Cord)
faults.330	ALARM	R/O	AlarmFlagRange R/O (Phase)
faults.340	ALARM	R/O	AlarmFlagRange R/O (Over-current prot. 01)
faults.341	ALARM	R/O	AlarmFlagRange R/O (Over-current prot. 02)
faults.350	ALARM	R/O	AlarmFlagRange R/O (Branch 01)
faults.351	ALARM	R/O	AlarmFlagRange R/O (Branch 02)
faults.93	ALARM	R/O	AlarmFlagRange R/O (Temp Sensor 1)
faults.94	ALARM	R/O	AlarmFlagRange R/O (Humidity Sensor 1)
faults.96	ALARM	R/O	AlarmFlagRange R/O (Temp Sensor 2)
faults.97	ALARM	R/O	AlarmFlagRange R/O (Humidity Sensor 2)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 Low Load)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 High Load)
faults.03	ALARM	R/O	AlarmFlagRange R/O (0utlet 1 Over Load)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 Low Load)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 High Load)
faults.06	ALARM	R/O	AlarmFlagRange R/O (0utlet 2 Over Load)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Outlet 3 Low Load)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Outlet 3 High Load)
faults.09	ALARM	R/O	AlarmFlagRange R/O (0utlet 3 Over Load)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 Low Load)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 High Load)
faults.12	ALARM	R/O	AlarmFlagRange R/O (0utlet 4 Over Load)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 Low Load)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 High Load)
faults.15	ALARM	R/O	AlarmFlagRange R/O (0utlet 5 Over Load)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Outlet 6 Low Load)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Outlet 6 High Load)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (0utlet 6 Over Load)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 Low Load)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 High Load)
faults.21	ALARM	R/O	AlarmFlagRange R/O (0utlet 7 Over Load)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 Low Load)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 High Load)
faults.24	ALARM	R/O	AlarmFlagRange R/O (0utlet 8 Over Load)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Outlet 9 Low Load)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Outlet 9 High Load)
faults.27	ALARM	R/O	AlarmFlagRange R/O (0utlet 9 Over Load)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 Low Load)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 High Load)
faults.30	ALARM	R/O	AlarmFlagRange R/O (0utlet 10 Over Load)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 Low Load)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 High Load)
faults.33	ALARM	R/O	AlarmFlagRange R/O (0utlet 11 Over Load)

name	*type*	*flags*	*range*
faults.34	ALARM	R/O	AlarmFlagRange R/O (Outlet 12 Low Load)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Outlet 12 High Load)
faults.36	ALARM	R/O	AlarmFlagRange R/O (0utlet 12 Over Load)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 Low Load)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 High Load)
faults.39	ALARM	R/O	AlarmFlagRange R/O (0utlet 13 Over Load)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 Low Load)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 High Load)
faults.42	ALARM	R/O	AlarmFlagRange R/O (0utlet 14 Over Load)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Outlet 15 Low Load)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Outlet 15 High Load)
faults.45	ALARM	R/O	AlarmFlagRange R/O (0utlet 15 Over Load)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 Low Load)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 High Load)
faults.48	ALARM	R/O	AlarmFlagRange R/O (0utlet 16 Over Load)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 Low Load)

name	*type*	*flags*	*range*
faults.50	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 High Load)
faults.51	ALARM	R/O	AlarmFlagRange R/O (0utlet 17 Over Load)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Outlet 18 Low Load)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Outlet 18 High Load)
faults.54	ALARM	R/O	AlarmFlagRange R/O (0utlet 18 Over Load)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 Low Load)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 High Load)
faults.57	ALARM	R/O	AlarmFlagRange R/O (0utlet 19 Over Load)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 Low Load)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 High Load)
faults.60	ALARM	R/O	AlarmFlagRange R/O (0utlet 20 Over Load)
faults.61	ALARM	R/O	AlarmFlagRange R/O (Outlet 21 Low Load)
faults.62	ALARM	R/O	AlarmFlagRange R/O (Outlet 21 High Load)
faults.63	ALARM	R/O	AlarmFlagRange R/O (0utlet 21 Over Load)
faults.64	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 Low Load)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 High Load)

name	*type*	*flags*	*range*
faults.66	ALARM	R/O	AlarmFlagRange R/O (0utlet 22 Over Load)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 Low Load)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 High Load)
faults.69	ALARM	R/O	AlarmFlagRange R/O (0utlet 23 Over Load)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Outlet 24 Low Load)
faults.71	ALARM	R/O	AlarmFlagRange R/O (Outlet 24 High Load)
faults.72	ALARM	R/O	AlarmFlagRange R/O (0utlet 24 Over Load)
faults.201	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 Warn)
faults.101	ALARM	R/O	AlarmFlagRange R/O (Outlet 1 Fault)
faults.202	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 Warn)
faults.102	ALARM	R/O	AlarmFlagRange R/O (Outlet 2 Fault)
faults.203	ALARM	R/O	AlarmFlagRange R/O (0utlet 3 Warn)
faults.103	ALARM	R/O	AlarmFlagRange R/O (0utlet 3 Fault)
faults.204	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 Warn)
faults.104	ALARM	R/O	AlarmFlagRange R/O (Outlet 4 Fault)
faults.205	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 Warn)

name	*type*	*flags*	*range*
faults.105	ALARM	R/O	AlarmFlagRange R/O (Outlet 5 Fault)
faults.206	ALARM	R/O	AlarmFlagRange R/O (0utlet 6 Warn)
faults.106	ALARM	R/O	AlarmFlagRange R/O (0utlet 6 Fault)
faults.207	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 Warn)
faults.107	ALARM	R/O	AlarmFlagRange R/O (Outlet 7 Fault)
faults.208	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 Warn)
faults.108	ALARM	R/O	AlarmFlagRange R/O (Outlet 8 Fault)
faults.209	ALARM	R/O	AlarmFlagRange R/O (0utlet 9 Warn)
faults.109	ALARM	R/O	AlarmFlagRange R/O (0utlet 9 Fault)
faults.210	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 Warn)
faults.110	ALARM	R/O	AlarmFlagRange R/O (Outlet 10 Fault)
faults.211	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 Warn)
faults.111	ALARM	R/O	AlarmFlagRange R/O (Outlet 11 Fault)
faults.212	ALARM	R/O	AlarmFlagRange R/O (0utlet 12 Warn)
faults.112	ALARM	R/O	AlarmFlagRange R/O (0utlet 12 Fault)
faults.213	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 Warn)

name	*type*	*flags*	*range*
faults.113	ALARM	R/O	AlarmFlagRange R/O (Outlet 13 Fault)
faults.214	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 Warn)
faults.114	ALARM	R/O	AlarmFlagRange R/O (Outlet 14 Fault)
faults.215	ALARM	R/O	AlarmFlagRange R/O (Outlet 15 Warn)
faults.115	ALARM	R/O	AlarmFlagRange R/O (0utlet 15 Fault)
faults.216	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 Warn)
faults.116	ALARM	R/O	AlarmFlagRange R/O (Outlet 16 Fault)
faults.217	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 Warn)
faults.117	ALARM	R/O	AlarmFlagRange R/O (Outlet 17 Fault)
faults.218	ALARM	R/O	AlarmFlagRange R/O (Outlet 18 Warn)
faults.118	ALARM	R/O	AlarmFlagRange R/O (0utlet 18 Fault)
faults.219	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 Warn)
faults.119	ALARM	R/O	AlarmFlagRange R/O (Outlet 19 Fault)
faults.220	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 Warn)
faults.120	ALARM	R/O	AlarmFlagRange R/O (Outlet 20 Fault)
faults.221	ALARM	R/O	AlarmFlagRange R/O (Outlet 21 Warn)

name	*type*	*flags*	*range*
faults.121	ALARM	R/O	AlarmFlagRange R/O (0utlet 21 Fault)
faults.222	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 Warn)
faults.122	ALARM	R/O	AlarmFlagRange R/O (Outlet 22 Fault)
faults.223	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 Warn)
faults.123	ALARM	R/O	AlarmFlagRange R/O (Outlet 23 Fault)
faults.224	ALARM	R/O	AlarmFlagRange R/O (Outlet 24 Warn)
faults.124	ALARM	R/O	AlarmFlagRange R/O (0utlet 24 Fault)

9.633 Show-Orbit

This page describes the device driver and the device window for the*Show-Orbit* logical device. This device calculates a satellite's orbit position from the azimuth angle of the antenna.

The *Show-Orbit* logical device may be used to display the orbit position an antenna is actually pointing to. Please note, that the device does not look at the antenna's elevation angle, the orbit position is calculated straight from the azimuth value.

There are two read-only variables representing the calculated orbit position:*orbit* contains the value as a floating point number expressed in degrees eastern longitude. This variable conforms to the standard notation of satellite positions in the sat-nms software. It may be used for both, displaying the value and feeding it into other calculations. The variable *orbitText* displays the value as a string with two digits precision and a "°E" or "°W" unit appended, depending on the satellites position.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the calculated satellite orbit positions.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	*description*
azimuthId	The parameter ID of the azimuth angle to monitor.
longitude	The longitude of the antenna's geodetic location.
latitude	The latitude of the antenna's geodetic location.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.azimuthId	TEXT	SAVE SETUP	StringRange
config.longitude	FLOAT	SAVE SETUP	DoubleRange (-180.00 .. 180.00)
config.latitude	FLOAT	SAVE SETUP	DoubleRange (-90.00 .. 90.00)
orbit	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
orbitText	TEXT	R/O	StringRange R/O

9.634 Sicilsat-S00761

This page describes the device driver and the device window for the Sicilsat S00761 Antenna Controller via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the antenna pointing parameters.
- --- Contains the target list
- --- Contains the tracking parameters.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.memTrackState	TEXT	R/O	StringRange R/O
internal.status	HEX	R/O	HexRange R/O (0 .. 0)
internal.track	INTEGER		IntegerRange (0 .. 0)
internal.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.bit14	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.bit15	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.bit16	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)

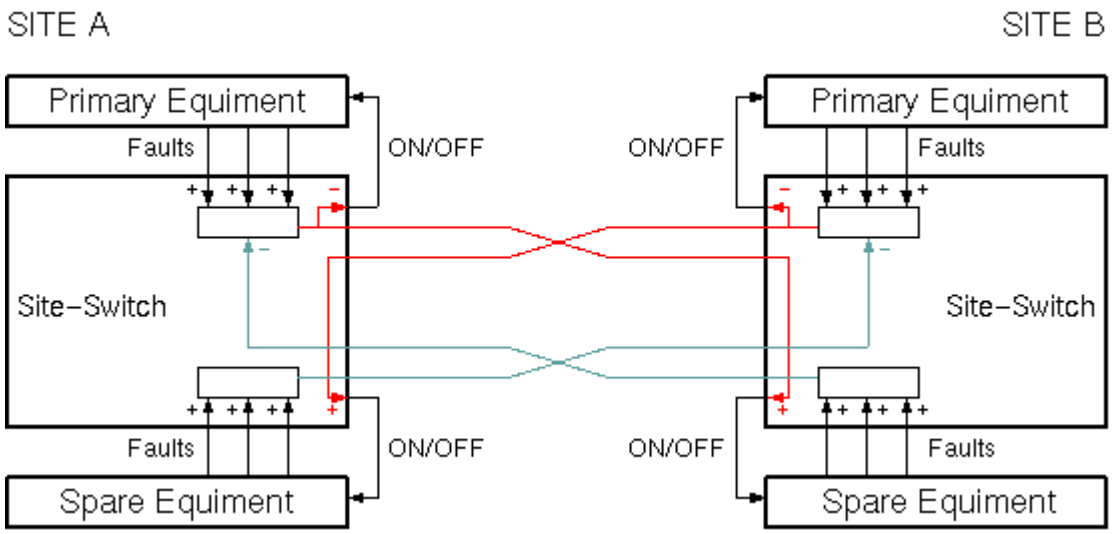
name	*type*	*flags*	*range*
step	TEXT	NOPRESET	StringRange
standby	TEXT	NOPRESET	StringRange
memory	TEXT	NOPRESET	StringRange
steptrack	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
trackMode	CHOICE	R/O	EnumRange R/O (STANDBY MANUAL PRESET STEP TRACK MEMORY TRACK)
stepSize	FLOAT		DoubleRange (0.005 .. 9.999)
restartLevel	FLOAT		DoubleRange (0.01 .. 9.99)
interval	INTEGER		IntegerRange (1 .. 60)
levelAveraging	INTEGER		IntegerRange (1 .. 999)
levelThreshold	FLOAT		DoubleRange (0.01 .. 9.99)
deltaError	FLOAT		DoubleRange (0.01 .. 9.99)
target.no	INTEGER		IntegerRange (0 .. 0)
target.name	TEXT		StringRange
target.list	TEXT	R/O	StringRange R/O
target.goto	INTEGER	NOPRESET	IntegerRange (2 .. 100)
goto	INTEGER	NOPRESET	IntegerRange (2 .. 100)
grmc	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CW)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (SW Limit AZ CCW)
faults.03	ALARM	R/O	AlarmFlagRange R/O (SW Limit El Up)
faults.04	ALARM	R/O	AlarmFlagRange R/O (SW Limit El Down)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HW Limit Az CW)
faults.06	ALARM	R/O	AlarmFlagRange R/O (HW Limit Az CCW)
faults.07	ALARM	R/O	AlarmFlagRange R/O (HW Limit Az Up)
faults.08	ALARM	R/O	AlarmFlagRange R/O (HW Limit Az Down)
faults.09	ALARM	R/O	AlarmFlagRange R/O (GPS Link)
faults.10	ALARM	R/O	AlarmFlagRange R/O (GPS Data)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low Beacon)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Sat Lock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Int. 5V PSU)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Int 12V PSU)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Remote Access)

9.635 Site-Switch

The Site-Switch logical device implements a site diversity switch control. Mostly used to backup up-link facilities at different sites, the Site-Switch device may be configured to work in various environments.

The primary equipment normally used for the up-link is monitored by the Site-Switch device. If the primary equipment fails, the local up-link gets shut down and the spare transmission facility at the remote site gets switched on. The control of the spare equipment at the remote site is done by a Site-Switch device, too. The site to site communication between the Site-Switch devices is performed via a TCP/IP network connection. The flow of signal ON/OFF commands between the sites appear as red lines in the figure below.



Due to the symmetrical design of the Site-Switch a setup like shown in the figure above can be done. The spare equipment at site B backs up the transmission from site A an vice versa. In a simpler configuration the spare equipment at site A and the primary equipment at site B do not apply.

Beside monitoring the primary equipment faults, the Site-Switch device also monitors the state of the spare equipment (in an asymmetrical setup this applies only to the Site-Switch device at site B). If the spare equipment at the redundant site fails, this is reported to the primary site and prevents the Site-Switch device there from switching to remote. The figure above shows this information as blue lines.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control page for the device. The operational parameters of the Site-Switch device are described below.
- --- The fault page:
  - **Remote Connection** : The TCP/IP connection to the remote site is broken.
  - **Local spare equipment** : The spare equipment at this location is not ready.
  - **Remote spare equipment** : The spare equipment at the remote site ist not ready.
  - **Latched condition** : A site switch already occurs.
- --- The device info page.
- --- The maintenance page. The configuration parameters described later are set at this page.

Operational parameters

The operational parameters of the Site-Switch device are set at the main control page. The table below explains each parameter in particular.

- *parameter --- description*
- actual location --- This parameter shows if the up-link is done at this location (LOCAL) or at the backup site (REMOTE). A manual switch over between the sites may be done at any time by changing this value. Be aware, that a switch over causes a transmission outage for a couple of seconds.
- switchover mode --- The device knows three different operating modes.
    - MANUAL --- No monitoring of primary equipment faults is done. The up-link location changes only on operator request.
    - 1-SHOT --- While the actual location is "LOCAL", the device monitors the primary equipment faults. If a fault is recognized, it switches over to "REMOTE" and reverts to MANUAL mode. Hence, after a switchover occurred the up-link location remains at the backup site until an operator changes this.
    - AUTOMATIC --- In AUTOMATIC mode, the device performs site switches in both directions. If the local equipment fails, it switches to remote. If the equipment is OK again, the up-link is switched back to the primary site. You should consider that this operating mode may cause the system to "oscillate", i.e. permanent site switches from A to B and back again may occur.
- switch delay --- Faults detected at the local (primary) equipment must continuously be present for the time defined with this parameter in order to trigger switchover. Set this time to prevent the Site-Switch from reporting to momentary faults
- recovery delay --- In AUTOMATIC mode, this parameter defines the time, the local (primary) equipment continuously must be OK until the Site-Switch initiates a switch back to the primary site. Setting this time to a larger value reduces the risk that the site switching oscillates.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Site-Switch device with the device id shall monitor/control.

parameter	***description***
spareFault1Id spareFault2Id spareFault3Id	You may define up to three message identifiers of device fault parameter which indicate that the spare equipment is faulty. The Site-Switch recognizes fault messages like "MODEM-1.fault" or "MODEM-1.faults.03". The spare equipment is assumed to be faulty if at least one of the faults is active.
primFault1Id primFault2Id primFault3Id	You may define up to three message identifiers of device fault parameter which indicate that the primary equipment is faulty. The Site-Switch recognizes fault messages like "MODEM-1.fault" or "MODEM-1.faults.03". The spare equipment is assumed to be faulty if at least one of the faults is active.

parameter	*description*
spareRfOnId	Define the identifier of the message the Site-Switch shall issue in order to switch the spare up-link on or off, i.e. "HPA-2.tx.on". The Site-Switch sends "ON" or "OFF" as parameter value with this message.
primRfOnId	Define the identifier of the message the Site-Switch shall issue in order to switch the primary up-link on or off, i.e. "HPA-2.tx.on". The Site-Switch sends "ON" or "OFF" as parameter value with this message.
remoteIpAddress	Specify the IP address of the remote M&C computer with this parameter, either in dotted quad notation or as a computer name the local machine is able to resolve.

Remarks

- The Site-Switch devices require a *permanent* network connection between the two sites. They are sensing the presence of this connection with a cyclic "ping" and will report a fault if the network link is down. If you are using an "on demand" dial-up link between the M&C computers, the ping probably will keep the link alive all the time.
- The Site-Switch provides a special fault flag called "Latched condition". This fault is set if the switch performs a switchover, it gets reset if the location is switched back *manually* . This is for use with the "1-SHOT" mode, the fault will tell the operator that a manual interaction is required. You may set the priority of this fault to "OFF" of you don't like the Site-Switch device to show a fault after a switchover occurred.
- If the simple OR relationship of the fault variables is not sufficient for your application, the Fault-Array logical device provides more sophisticated capabilities to link faults and other parameter values together. You may route a Fault-Array in between the fault sources and the Site-Switch device.
- A the output the Site-Switch may extended using aSwitch-Cross-Link logical device. This device is capable in translating the ON/OFF command into waveguide switch settings, duplicating the commands to multiple devices or to adjust transmission levels as a result of the switchover.
- The delay times of the fault recognition are implemented by the background polling procedure which is called on a regular basis for all logical devices. With the factory preset settings, this procedure runs once a second. The resolution of the delay time is limited to this value.If your application needs a finer resolution, you may configure the polling procedures of all logical device to be called more frequently. This is done by configuring the *IDLE* parameter of the *PORT null* in the vlc.setup file. Be aware, that the workload of the M&C computer will rise dramatically if you set this parameter to very small values.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.spareFault1Id	TEXT	SAVE SETUP	StringRange
config.spareFault2Id	TEXT	SAVE SETUP	StringRange
config.spareFault3Id	TEXT	SAVE SETUP	StringRange
config.primFault1Id	TEXT	SAVE SETUP	StringRange
config.primFault2Id	TEXT	SAVE SETUP	StringRange
config.primFault3Id	TEXT	SAVE SETUP	StringRange
config.spareRfOnId	TEXT	SAVE SETUP	StringRange
config.primRfOnId	TEXT	SAVE SETUP	StringRange
config.remoteIpAddress	TEXT	SAVE SETUP	StringRange
location	CHOICE	SAVE	EnumRange (LOCAL REMOTE)
spare	CHOICE	SAVE	EnumRange (OFF ON)
inhibit	BOOLEAN		BooleanRange (true false)
switchMode	CHOICE	SAVE	EnumRange (MANUAL 1-SHOT AUTOMATIC)
lrDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
rlDelay	FLOAT	SAVE	DoubleRange (0.0 .. 300.0)
updateInhibit	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote connection)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote spare equip.)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Local spare equip.)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Latched condition)

9.636 Spacepath-CM2-TWTA

This page describes the device driver and the device window for the Spacepath TWTA over SNMP using the CM2 MIB. It doesn't support redundancy parameters.

It was tested with Model STA 5375

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	**description**
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
lowPowerAlarm	Sets the low rf output alarm limit in dBm

parameter	description
lowPowerFault	Sets the low rf output faults limit in dBm
highPowerAlarm	Sets the high rf output alarm limit in dBm
highPowerFault	Sets the high rf output fault limit in dBm
upc.source	Reference signal source for built-in Uplink-Power-Control
config.upc.faultAction	Action to be performed on UPC fault
logGainChanges	Defines if gain changes should be written in the event log database.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange

name	*type*	*flags*	*range*
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.tx.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.supply15v	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.supply24v	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.boostVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.fanVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.redundancyMode	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowPowerAlarm	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.lowPowerFault	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.highPowerAlarm	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.highPowerFault	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.upc.source	CHOICE	SETUP	EnumRange (THIS AMPLIFIER HPA-A HPA-B HPA-C CONTROLLER)

name	*type*	*flags*	*range*
config.upc.faultAction	CHOICE	SETUP	EnumRange (CLEARSKY-PWR LAST-PWR)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
state	TEXT	R/O	StringRange R/O
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.00 .. 30.00)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
heaterDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
heaterState	TEXT	R/O	StringRange R/O
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uHigh	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.uAnode	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.iBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.twtTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.alcState	TEXT	R/O	StringRange R/O
tx.alc	CHOICE		EnumRange (OFF ALC UPC GTP)
tx.requestedLevel	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
upc.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
upc.manualRatio	FLOAT		DoubleRange (0.1 .. 60.0)
upc.clearSkyVoltage	FLOAT		DoubleRange (0.00 .. 24.00)
upc.clearSkyPower	FLOAT		DoubleRange (0.00 .. 60.00)
clearFaults	TEXT	NOPRESET	StringRange
enableRemote	TEXT	NOPRESET	StringRange
internal.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.refPwr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU current)
faults.03	ALARM	R/O	AlarmFlagRange R/O (HV range)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HV under voltage)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HV over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Beam current)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Mean helix current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Peak helix current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (HV arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Heater current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Heater voltage)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Negative anode)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (Positive anode)
faults.14	ALARM	R/O	AlarmFlagRange R/O (15V)
faults.15	ALARM	R/O	AlarmFlagRange R/O (BUC)
faults.16	ALARM	R/O	AlarmFlagRange R/O (ACM temperature)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PPM temperature)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RFM Comms)
faults.20	ALARM	R/O	AlarmFlagRange R/O (FM comms)
faults.21	ALARM	R/O	AlarmFlagRange R/O (BUC comms)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Voltage boost)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Clock)
faults.24	ALARM	R/O	AlarmFlagRange R/O (TWT thermal switch)
faults.25	ALARM	R/O	AlarmFlagRange R/O (CPU temperature)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (PIC processor)
faults.28	ALARM	R/O	AlarmFlagRange R/O (RFM 24V)
faults.29	ALARM	R/O	AlarmFlagRange R/O (RFD temperature)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Attenuator temperature)
faults.31	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Low RF)

name	*type*	*flags*	*range*
faults.33	ALARM	R/O	AlarmFlagRange R/O (High Reflected Power)
faults.34	ALARM	R/O	AlarmFlagRange R/O (RF arc 1)
faults.35	ALARM	R/O	AlarmFlagRange R/O (RF arc 2)
faults.36	ALARM	R/O	AlarmFlagRange R/O (BUC)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Lock loss)
faults.38	ALARM	R/O	AlarmFlagRange R/O (FM 24V)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Fan voltage)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.41	ALARM	R/O	AlarmFlagRange R/O (TWT temperature)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Interlock 1)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Interlock 2)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Interlock 3)
faults.45	ALARM	R/O	AlarmFlagRange R/O (ALC UPC)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Fault Reset Disabled)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.48	ALARM	R/O	AlarmFlagRange R/O (External Inhibit 1)
faults.49	ALARM	R/O	AlarmFlagRange R/O (External Inhibit 2)
faults.50	ALARM	R/O	AlarmFlagRange R/O (External Inhibit 3)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Chassis interlock)
faults.52	ALARM	R/O	AlarmFlagRange R/O (SD Card)

9.637 Spacepath-TWTA-Serial

This page describes the device driver and the device window for the Spacepath STA TWTA via Serial Protocol. It uses the Standard command set. For models supporting SNMP it is recommended to use the driver: Spacepath-TWTA

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- lowLimitOn --- Defines the low rf output power limit behaviour
  - OFF - No Monitoring
  - TRIP - Low RF Alarm and switch off
  - WARNING - Low RF Warning
- highLimitOn --- Defines the high rf power limit behaviour

    - OFF - No Monitoring
    - TRIP - High RF Alarm and switch off
    - WARNING - High RF Warning

- lowLimit --- Defines the low rf output limit in W
- highLimit --- Defines the high rf output limit in W
- logGainChanges --- Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
- remoteMode --- Defines the remote control mode

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.bootLoader	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.standbyElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.totalElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.offElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lowLimitOn	CHOICE	R/O	EnumRange R/O (OFF ON)
config.highLimitOn	CHOICE	R/O	EnumRange R/O (OFF ON)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.remoteMode	CHOICE	SETUP	EnumRange (REMOTE LOCAL)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
state	CHOICE	R/O	EnumRange R/O (OFF WARMUP STANDBY BEAM ON FAILURE)
attn.integer	INTEGER		IntegerRange (0 .. 4095)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
heaterDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uCollector	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uCathode	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.twtTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PFC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (DCB)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HV Overload)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mean Helix Current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Peak Helix Current)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Case Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (TWT Temperature)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Stirrer)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cooler Fan)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TWT Cathode Voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (TWT Collector Voltage)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT Heater Voltage)
faults.17	ALARM	R/O	AlarmFlagRange R/O (External Reference)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Upconverter 10MHz)
faults.19	ALARM	R/O	AlarmFlagRange R/O (ALC Input High)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (ALC Input Low)
faults.21	ALARM	R/O	AlarmFlagRange R/O (High RF Warn)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low RF Warn)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Chassis Interlock)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.638 Spacepath-TWTA

This page describes the device driver and the device window for the Spacepath TWTA over SNMP. It doesn't support redundancy parameters.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- Displays the redundancy settings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
    - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.
- lowLimitOn --- Defines the low rf output power limit behaviour
  - OFF - No Monitoring
  - TRIP - Low RF Alarm and switch off
  - WARNING - Low RF Warning
- highLimitOn --- Defines the high rf power limit behaviour
  - OFF - No Monitoring
  - TRIP - High RF Alarm and switch off
  - WARNING - High RF Warning
- lowLimit --- Defines the low rf output limit in W
- highLimit --- Defines the high rf output limit in W
- linearizer --- Switch on or off the integrated linearizer
- linearizerMagn --- Sets the linearizer magnitude as an integer 0-255
- linearizerPhase --- Sets the linearizer phase as an integer 0-255
- linearizerGain --- Sets the linearizer gain as an integer 0-255
- remoteMode --- Defines the remote control mode
- ctrlMode --- Defines the control mode. For usage with the satnms system only ALL and SNMP are suitable

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.bucInstalled	CHOICE	R/O	EnumRange R/O (NO YES)
info.linInstalled	CHOICE	R/O	EnumRange R/O (NO YES)
info.linXInstalled	CHOICE	R/O	EnumRange R/O (NO YES)
info.attnInstalled	CHOICE	R/O	EnumRange R/O (NO YES)
info.model	TEXT	R/O	StringRange R/O
info.bands	CHOICE	R/O	EnumRange R/O (NONE C-BAND X-BAND KU-BAND DBS-BAND C+X+KU-BAND C+X-BAND C+KU-BAND KU+DBS-BAND C+X+KU+DBS-BAND)
info.serial	TEXT	R/O	StringRange R/O
info.bootLoader	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.maxAttn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.tripCounter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.standbyElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.totalElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.lowLimitOn	CHOICE	SETUP	EnumRange (OFF TRIP WARNING)
config.highLimitOn	CHOICE	SETUP	EnumRange (OFF TRIP WARNING)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 1000)
config.linearizer	CHOICE	SETUP	EnumRange DIS (ON OFF)
config.linearizerMagn	INTEGER	SETUP	IntegerRange DIS (0 .. 255)
config.linearizerPhase	INTEGER	SETUP	IntegerRange DIS (0 .. 255)
config.linearizerGain	INTEGER	SETUP	IntegerRange DIS (0 .. 255)
config.remoteMode	CHOICE	SETUP	EnumRange (LOCAL REMOTE)
config.ctrlMode	CHOICE	SETUP	EnumRange (NONE ALL PANEL COM1 COM2 CLI TCP SNMP WEB RAW 9)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF STANDBY ON)
state	CHOICE	R/O	EnumRange R/O (OFF WARMUP STANDBY BEAM ON FAILURE)
tx.gain	FLOAT		DoubleRange DIS (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange DIS (0.0 .. 20.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
heaterDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uAnode	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uCollector	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uCathode	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.twtTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.alcState	CHOICE	R/O	EnumRange R/O (OFF INACTIVE ACTIVE INCREASE DECREASE INPUT LOW INPUT HIGH)
tx.alc	CHOICE		EnumRange (OFF ALC GTP)
tx.requestedLevel	FLOAT		DoubleRange (0.0 .. 0.0)
tx.alcRate	FLOAT		DoubleRange (0.00 .. 0.00)
clearFaultCount	TEXT	NOPRESET	StringRange
clearFaultRec	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
internal.model1	TEXT	R/O	StringRange R/O
internal.model2	TEXT	R/O	StringRange R/O
internal.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.requestedLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PFC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Arc detetcor)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HV Overload)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Mean Helix Current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Peak Helix Current)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Case Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (TWT Temperature)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Stirrer Fans)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cooling Fans)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TWT Cathode Voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (TWT Collector Voltage)
faults.16	ALARM	R/O	AlarmFlagRange R/O (TWT Heater Voltage)
faults.17	ALARM	R/O	AlarmFlagRange R/O (External Reference)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.19	ALARM	R/O	AlarmFlagRange R/O (ALC Input High)
faults.20	ALARM	R/O	AlarmFlagRange R/O (ALC Input Low)
faults.21	ALARM	R/O	AlarmFlagRange R/O (High RF Warn)

name	*type*	*flags*	*range*
faults.22	ALARM	R/O	AlarmFlagRange R/O (Low RF Warn)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Test)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Chassis Interlock)
faults.24	ALARM	R/O	AlarmFlagRange R/O (HW Mode Remote)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Control Mode)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.639 Spacepath-Tango-TWTA

This page describes the device driver and the device window for the Spacepath TWTA over SNMP using the Tango MIB. It doesn't support redundancy parameters.

It was tested with Model STA 5340

- Firmware version 21

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The transmit parameters
- --- Displays meter readings like output power and BUC specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	**description**

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
maxAttn	Defines units max attenuation. Default setting is 50dB
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
lowPowerAlarm	Sets the low rf output alarm limit in dBm
lowPowerFault	Sets the low rf output faults limit in dBm
highPowerAlarm	Sets the high rf output alarm limit in dBm
highPowerFault	Sets the high rf output fault limit in dBm
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.cmVersion	TEXT	R/O	StringRange R/O
info.fmVersion	TEXT	R/O	StringRange R/O
info.rfmVersion	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.maxAttn	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowPowerAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.lowPowerFault	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.highPowerAlarm	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.highPowerFault	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF ON)
state	TEXT	R/O	StringRange R/O
tx.gain	FLOAT		DoubleRange (0.0 .. 50.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 50.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uFan	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.alc	CHOICE		EnumRange (OFF ON)
tx.alcState	TEXT	R/O	StringRange R/O
tx.requestedLevel	FLOAT		DoubleRange (0.00 .. 0.00)
reset	TEXT	NOPRESET	StringRange
internal.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.date	TEXT	R/O	StringRange R/O
internal.gain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.requestedLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.faultString	TEXT	R/O	StringRange R/O
internal.faults.02	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.03	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.04	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.05	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.06	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.07	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.08	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.09	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.10	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.11	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.12	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.13	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.14	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.15	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.16	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.17	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.18	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.19	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.20	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.21	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.22	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.faults.23	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.24	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.25	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.27	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.28	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.29	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.30	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.31	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.32	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.33	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.34	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.35	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.36	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.37	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.38	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.39	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.40	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.41	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.42	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.43	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.44	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.45	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.46	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.47	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.48	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.faults.49	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.50	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.51	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.52	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.53	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.54	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.55	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.56	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.57	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.58	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.59	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.60	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.61	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.62	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.63	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.64	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.65	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.66	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.67	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.68	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.69	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.70	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.71	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.72	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.73	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (TWT thermal switch)
faults.03	ALARM	R/O	AlarmFlagRange R/O (BUC fault indicate)
faults.04	ALARM	R/O	AlarmFlagRange R/O (BUC loss of lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (BUC RF inhibit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan voltage)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Negative anode voltage)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Positive anode voltage)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Beacon receiver)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Beam current)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Helix current Warning)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Helix current)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Heater current)
faults.14	ALARM	R/O	AlarmFlagRange R/O (ACM PSU Temp. Warning)
faults.15	ALARM	R/O	AlarmFlagRange R/O (PPM PSU Temp. Warning)
faults.16	ALARM	R/O	AlarmFlagRange R/O (ACM PSU Temp.)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PPM PSU Temp.)
faults.18	ALARM	R/O	AlarmFlagRange R/O (TWT Temp)

name	*type*	*flags*	*range*
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF Driver Temp)
faults.20	ALARM	R/O	AlarmFlagRange R/O (RF Attn Temp)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Waveguide switch position)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Customer setup)
faults.25	ALARM	R/O	AlarmFlagRange R/O (AUX/M&C PSU Overcurrent)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Disc 1 Overcurrent)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Disc 2 Overcurrent)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Disc 3 Overcurrent)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Disc 1 external interlock)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Disc 2 external interlock)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Disc 3 external interlock)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Disc 1 external RF inhibit)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Disc 2 external RF inhibit)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Disc 3 external RF inhibit)
faults.36	ALARM	R/O	AlarmFlagRange R/O (High helix current)

name	*type*	*flags*	*range*
faults.37	ALARM	R/O	AlarmFlagRange R/O (Beam voltage exceeds limits)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Internal PSU current)
faults.39	ALARM	R/O	AlarmFlagRange R/O (High voltage arc)
faults.40	ALARM	R/O	AlarmFlagRange R/O (High voltage set point)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Reset Attempts)
faults.42	ALARM	R/O	AlarmFlagRange R/O (RF High)
faults.43	ALARM	R/O	AlarmFlagRange R/O (RF Low)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Reflected power A)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Reflected power B)
faults.46	ALARM	R/O	AlarmFlagRange R/O (RF arc amp)
faults.47	ALARM	R/O	AlarmFlagRange R/O (RF arc system)
faults.48	ALARM	R/O	AlarmFlagRange R/O (BUC loss of lock)
faults.49	ALARM	R/O	AlarmFlagRange R/O (BUC fault)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Unit A muted)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Unit B muted)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Unit C muted)
faults.53	ALARM	R/O	AlarmFlagRange R/O (System muted via unit A)
faults.54	ALARM	R/O	AlarmFlagRange R/O (System muted via unit B)

name	*type*	*flags*	*range*
faults.55	ALARM	R/O	AlarmFlagRange R/O (System muted via unit C)
faults.56	ALARM	R/O	AlarmFlagRange R/O (RFM not online RF Inhibit)
faults.57	ALARM	R/O	AlarmFlagRange R/O (RFM 24V RF Inhibit)
faults.58	ALARM	R/O	AlarmFlagRange R/O (RFM Temp. RF Inhibit)
faults.59	ALARM	R/O	AlarmFlagRange R/O (RFM BUC RF Inhibit)
faults.60	ALARM	R/O	AlarmFlagRange R/O (RFM Amplifier Arc 1 RF Inhibit)
faults.61	ALARM	R/O	AlarmFlagRange R/O (RFM System Arc RF Inhibit)
faults.62	ALARM	R/O	AlarmFlagRange R/O (RFM exceeded RF Power RF Inhibit)
faults.63	ALARM	R/O	AlarmFlagRange R/O (BUC unlocked)
faults.64	ALARM	R/O	AlarmFlagRange R/O (BUC communication)
faults.65	ALARM	R/O	AlarmFlagRange R/O (BUC int/ext reference)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Redundancy)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Backup not ready)
faults.68	ALARM	R/O	AlarmFlagRange R/O (FM not communicating)
faults.69	ALARM	R/O	AlarmFlagRange R/O (RFM not communicating)
faults.70	ALARM	R/O	AlarmFlagRange R/O (Incomp FM version)

name	*type*	*flags*	*range*
faults.71	ALARM	R/O	AlarmFlagRange R/O (Incomp RFM version)
faults.72	ALARM	R/O	AlarmFlagRange R/O (RFM user inhibit)
faults.73	ALARM	R/O	AlarmFlagRange R/O (RF inhibit)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.640 Speciality-BDC-Controller

This page describes the device driver and the device window for the 2 x 1:1 BDC Redundant System Controllers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The Speciality do not need any configuration parameters except the settings for the serial interface. The default address is /.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *9600,N,7,1* .

Remote Control

- Default TCP port 6001
- Default Address "/"
- TCP uses Terminal LF protocol

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.address	TEXT	SAVE SETUP	StringRange
protection.1	CHOICE		EnumRange (MANUAL AUTO)
protection.2	CHOICE		EnumRange (MANUAL AUTO)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unit 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Unit 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protection 1/2)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection 3/4)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
internal.state	TEXT	R/O	StringRange R/O
internal.w	HEX	R/O	HexRange R/O (0 .. F)
internal.x	HEX	R/O	HexRange R/O (0 .. F)
internal.y	HEX	R/O	HexRange R/O (0 .. F)
internal.z	HEX	R/O	HexRange R/O (0 .. F)
internal.nw	HEX	R/O	HexRange R/O (0 .. F)
internal.nx	HEX	R/O	HexRange R/O (0 .. F)
internal.ny	HEX	R/O	HexRange R/O (0 .. F)
internal.nz	HEX	R/O	HexRange R/O (0 .. F)
internal.s1p1	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.s1p2	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.s2p1	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.s2p2	INTEGER	R/O	IntegerRange R/O (0 .. 1)
internal.s1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.s2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.641 Speciality-Controller-1926-100

This page describes the device driver and the device window for the customized Speciality Microwave controller 1926-100 via SMCBus protocol, The Waveguide Switching and HPA Protection system is tested with up to 6 waveguide switches and 3+1 HPA redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address as set at the device as Decimal. Default address = 1
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
protection	CHOICE		EnumRange (AUTO MANUAL)
switch.01	CHOICE		EnumRange (A B)
switch.02	CHOICE		EnumRange (A B)
switch.03	CHOICE		EnumRange (A B)
switch.04	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
switch.05	CHOICE		EnumRange (A B)
switch.06	CHOICE		EnumRange (A B)
reset	TEXT	NOPRESET	StringRange
internal.f1	HEX	R/O	HexRange R/O (0 .. 0)
internal.f2	HEX	R/O	HexRange R/O (0 .. 0)
internal.f3	HEX	R/O	HexRange R/O (0 .. 0)
internal.f4	HEX	R/O	HexRange R/O (0 .. 0)
internal.prot	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.local1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.local2	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (TWTA 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (TWTA 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TWTA 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TWTA 4)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Switch 6)

9.642 Speciality-Controller-9753-500

This page describes the device driver and the device window for the customized Speciality Microwave controller 9753-500 via SMCBus protocol, The Waveguide Switching and HPA Protection system is tested with up to 8 waveguide switches 1+1 KPA and 2+1 HPA redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address as set at the device as ASCII-character.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
protection	CHOICE		EnumRange (AUTO MANUAL)
switch.01	CHOICE		EnumRange (A B)
switch.02	CHOICE		EnumRange (A B)
switch.03	CHOICE		EnumRange (A B)
switch.04	CHOICE		EnumRange (A B)
switch.05	CHOICE		EnumRange (A B)
switch.06	CHOICE		EnumRange (A B)
switch.07	CHOICE		EnumRange (A B)
switch.08	CHOICE		EnumRange (A B)
reset	TEXT	NOPRESET	StringRange
internal.f1	HEX	R/O	HexRange R/O (0 .. 0)
internal.f2	HEX	R/O	HexRange R/O (0 .. 0)
internal.f3	HEX	R/O	HexRange R/O (0 .. 0)
internal.f4	HEX	R/O	HexRange R/O (0 .. 0)
internal.prot	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.local1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.local2	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TWTA Local)
faults.04	ALARM	R/O	AlarmFlagRange R/O (KPA Local)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Protection)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (TWTA 1A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (TWTA 2A)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TWTA 3A)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TWTA 1B)
faults.22	ALARM	R/O	AlarmFlagRange R/O (TWTA 2B)
faults.23	ALARM	R/O	AlarmFlagRange R/O (TWTA 3B)
faults.31	ALARM	R/O	AlarmFlagRange R/O (KPA 1)
faults.32	ALARM	R/O	AlarmFlagRange R/O (KPA 2)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Switch 1)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Switch 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Switch 3)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Switch 4)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Switch 5)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Switch 6)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Switch 7)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Switch 8)

9.643 Spectrum-QAccess

This page describes the device driver and the device window for the*Spectrum-QAccess* logical device. This device monitors a number of parameters at a receiving device (e.g. IRD, Modem or IRD-Chain) and on demand sets the parameters on a spectrum analyzer to view the received signal.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page only presents the "TAKE" button which sets the spectrum analyzer to the appropriate settings
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
frequencyId	The message ID of the receive frequency to monitor.
dataRateId	The message ID of the symbol data to monitor
inputSelectionId	The message ID of the receiver input selection to monitor.
inputDefault	The spectrum analyzer input switch setting to be used if no receiver input selection is monitored.
inputSwitchId	The message ID of the spectrum analyzer input switch position.
translateIn	Comma separated list of receiver input selection settings to translate.
translateOut	Comma separated list of spectrum analyzer input switch positions, corresponding to the receiver input selections above.
spectrumAnalyzer	Device name of the spectrum analyzer to control.
dataRateRatio	Symbol rate to frequency span ratio, see below.
instrumentPreset	If set to YES, the device first commands an "instrument preset" at the spectrum analyzer before center frequency an span are set.
delay	(0..10000msec), defines the delay between the input select command an the subsequent commands. The default delay is 1500msec which should be sufficient to get the correct LO frequency before the center frequency is set, but in some situations it may be necessary to increase this value.

Spectrum analyzer input switch settings

The Spectrum-QAccess device controls an N:1 switch in front of the spectrum analyzer. There are two configurations which the device can handle:

1. **The receiver uses an input signal selector or is connected to the** **antenna(s) through a switch matrix.**In this case enter the parameter ID of the receiver's input selector position at the "inputSelectionId" configuration parameter. If the receiver's input selections do not match the spectrum analyzer's input switch positions, list all possible receiver input selections in the "translateIn" configuration parameter, the corresponding spectrum analyzer input switch settings in the "translateOut" configuration parameter.When the device receives a "take" command, it reads the actual receiver input selection, translates it through the translateIn/translateOut lists and finally sets the

spectrum analyzer input switch to the evaluated position.If receiver and spectrum analyzer are connected to the same switch matrix, the input selections for both devices are identical. In this case you can leave the translateIn/translateOut lists empty, the device will pass through the switch position without change.

2. **The receiver is connected directly to the antenna port / signal** **distributor without a switch.**There is no receiver input selection to monitor, hence leave the "inputSelectionId", "translateIn" and "translateOut" parameters empty. Instead, enter the spectrum analyzer input switch position to set for this receiver in the "inputDefault" field.Please note, that if the device was configured to listen to a receiver input switch before, you need to restart the M&C software in order make the change effective.

Frequency span calculation

The device monitors the symbol rate setting of the receiver to evaluate the frequency span which is set at the spectrum analyzer. The span is calculated following the equation:

frequency-span = symbol-rate / dataRateRatio

When setting the dataRateRatio, you should take notice of the unit multipliers of the parameters involved:

The spectrum analyzer accepts the frequency span setting in MHz. DVB receivers usually show the symbol rate setting in Msym/sec, hence a dataRateRatio value of 0.5 will give reasonable results.

Data modems however provide the symbol rate setting in symbols / second. In this case the dataRateRatio value should be set to e.g. 500000 in order to scale the result to MHz.

Center frequency setting

The Spectrum-QAccess device copies the monitored receive frequency without change to the spectrum analyzer's center frequency setting. This generally works fine as the sat-nms software always displays these frequency values as MHz.

Care should be taken to apply the correct LO frequency to the spectrum analyzer input. The CSM-Spectrum-Analyzer provides a flexible method to derive the LO frequency from the input switch position.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.frequencyId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.dataRateId	TEXT	SAVE SETUP	StringRange
config.inputSelectionId	TEXT	SAVE SETUP	StringRange
config.inputDefault	TEXT	SAVE SETUP	StringRange
config.inputSwitchId	TEXT	SAVE SETUP	StringRange
config.translateIn	TEXT	SAVE SETUP	StringRange
config.translateOut	TEXT	SAVE SETUP	StringRange
config.spectrumAnalyzer	TEXT	SAVE SETUP	StringRange
config.dataRateRatio	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.instrumentPreset	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.delay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
take	TEXT		StringRange

9.644 Store-Float

The logical *Store-Float* device is a dummy device driver which stores a Float variable when triggered externally. It's value can be used e.g. as input parameter to trigger an Replicator device to make sure that Replication is only forwarded in one direction.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the stored float value.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

The device does not contain any configuration parameters.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
value	FLOAT		DoubleRange (0.000 .. 0.000)

9.645 Stream-KeySelect

The Stream-KeySelect logical device lets you comfortably set BISS encryption keys and streaming protocol passphrases from a central database. You may store the key /passphrase with a user defined description in the database and retrieve them by this descriptive text.

Stream-KeySelect reads its key definitions from the satellite database (SQL database). It replaces the older BISS-KeySelect-CDB device and adds support for streaming passphrases.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (key select) page for the device.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation modes

The Stream-KeySelect device provides different operating modes for BISS key and streaming passphrase management. By default the device resembles the operation mode of the older BISS-KeySelect-CDB device:

- The device duplicates the BISS-1/BISS-E selection parameter of the decoder / encoder device. The BISS mode may be selected either at the Stream-KeySelect device or at the decoder / encoder device itself.

- The list of available BISS keys from the database only contains the entries which match the actually selected BISS mode.
- Manually entered BISS keys are checked to be a valid hexadeximal number with 12 or 16 digits, depending on the selected BISS mode.

Alternatively the device may be used to manage stream passphrases stored in the database. In this mode the device reads the so called 'passmode' from the decoder / encoder device it controls. This passmode is a keyword which defines the destination parameter ID for a passphrase in this particular mode. This indirection has been introduced to cope with devices which expect the passphrase at different parameters for each IP protocol mode.

- To use the 'passmode' operation mode of the Stream-KeySelect device, the 'passmodeId' configuration parameter must be set to a variable at the decoder / encoder which reports the name of the passmode actually to use.
- For all known passmode names the controlled device may report, the passmode name and the corresponding destination for the passphrase must be configured.
- In 'passmode' operation mode the Stream-KeySelect device does not monitor or set a BISS-mode and it accepts a free text for manually entered passphrases.
- The Stream-KeySelect device may dynamically switch between BISS mode and one of the passphrase modes. If the parameter addressed by 'passmodeId' reads the predefined value 'BISS', the Stream-KeySelect device switches to BISS mode.

Operation

To set a Stream key at the device ion BISS mode, first select the BISS mode (BISS-E, BISS-1, ...). Then select the key by its name from the list of keys and press enter.

Alternatively you may enter a 12 or 16 digit hexadecimal number to the field, the device treats this as a BISS key to be forwarded without using the database. This way you easily can use encryption keys which are not stored in the database.

Most IRD devices do not allow to read back the BISS encryption key. The BISS-KeySelect-CDB device defangs this restriction by leaving the last key applied in the entry field. You should be aware, that there may be situations where the displayed key is not that one actually active in the IRD.

In the passphrase modes the field for the BISS mode is not applicable and greyed out. Simply select the name of the passphrase to apply and press enter. Alternatively you may enter a passphrase directly in the selection field, followed by 'Enter'.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Stream-KeySelect device with the equipment.

parameter	***description***

parameter	*description*
passmodeId	The ID for a parameter the Stream-KeySelect device uses to evaluate its operation mode (BISS, SRT etc..). Leave this field empty if the Stream-KeySelect controls an IRD which supports only BISS
passmode.1	The name / keyword for a passphrase destination as read from the parameter addressed by passmodeId. If the value of the passmodeId parameter matches the keyword configured here, any selected passphrase is sent to the parameter addressed by passphraseId.1 configuration parameter.
passphraseId.1	The parameter ID to send a passphrase to if the value addressed by the passmodeId parameter matches the value of passmode.1
passmode.2 .. 10	There are 10 passmode.# and passphrase.# pairs in total. The Stream-KeySelect device always checks all 10 passmode.# settings, if more than one match the actual passmode, all of the corresponding passphraseId.# parameters will receive a passphrase selected from the database.
passphraseId.2 .. 10	The destination parameter IDs to send a passphrase to if the passmode.# value with the same index matches the actual passmode.
bissmodeId	The message ID to read and set the BISS mode at the device.
bisskeyId	The message ID to set the encryption key at the device. If the device requires the key to be split in a lower and an upper part, leave this field empty and use the two fields below.
bisskeyLoId	The message ID to set the lower part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.
bisskeyHiId	The message ID to set the higher part of the encryption key. This is only required if the device (some encoders do so) uses two separate parameters for a split encryption key.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.passmode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.passmodeId	TEXT	SAVE SETUP	StringRange
config.passmode.1	TEXT	SAVE SETUP	StringRange
config.passphraseId.1	TEXT	SAVE SETUP	StringRange
config.passmode.2	TEXT	SAVE SETUP	StringRange
config.passphraseId.2	TEXT	SAVE SETUP	StringRange
config.passmode.3	TEXT	SAVE SETUP	StringRange
config.passphraseId.3	TEXT	SAVE SETUP	StringRange
config.passmode.4	TEXT	SAVE SETUP	StringRange
config.passphraseId.4	TEXT	SAVE SETUP	StringRange
config.passmode.5	TEXT	SAVE SETUP	StringRange
config.passphraseId.5	TEXT	SAVE SETUP	StringRange
config.passmode.6	TEXT	SAVE SETUP	StringRange
config.passphraseId.6	TEXT	SAVE SETUP	StringRange
config.passmode.7	TEXT	SAVE SETUP	StringRange
config.passphraseId.7	TEXT	SAVE SETUP	StringRange
config.passmode.8	TEXT	SAVE SETUP	StringRange

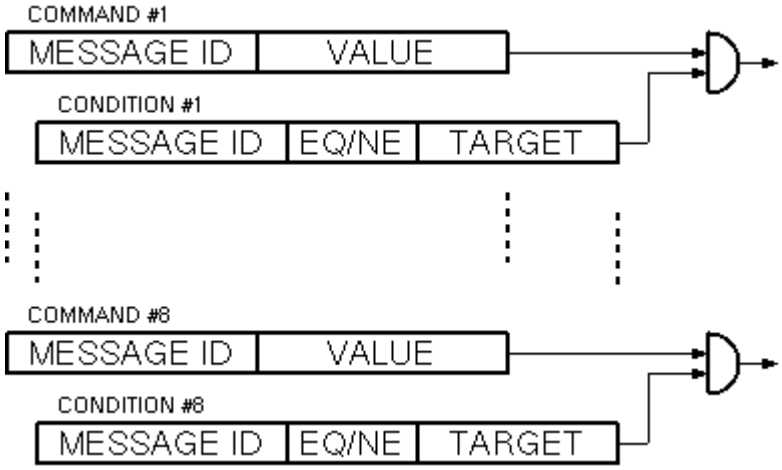
name	*type*	*flags*	*range*
config.passphraseId.8	TEXT	SAVE SETUP	StringRange
config.passmode.9	TEXT	SAVE SETUP	StringRange
config.passphraseId.9	TEXT	SAVE SETUP	StringRange
config.passmode.10	TEXT	SAVE SETUP	StringRange
config.passphraseId.10	TEXT	SAVE SETUP	StringRange
config.bissmodeId	TEXT	SAVE SETUP	StringRange
config.bisskeyId	TEXT	SAVE SETUP	StringRange
config.bisskeyLoId	TEXT	SAVE SETUP	StringRange
config.bisskeyHiId	TEXT	SAVE SETUP	StringRange
key	CHOICE		EnumRange (unknown)
bissMode	CHOICE		EnumRange (unknown)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.646 Switch-Cross-Link

The Switch-Cross-Link logical device translates a ON/OFF command into a set of arbitrary messages / commands. Individual commands may be bound to a condition, i.e. the command is sent only if another parameter matches (or matches not) a given target value. The Switch-Cross-Link device mostly is used to perform carrier ON/OFF switching in an environment where this cannot be simply by switching one device on or off.

The device stores two tables of eight commands each. The commands defined on the "ON" table get issued if the Switch-Cross-Link device receives an "ON" command, the commands

defined in the second table are sent if the device receives "OFF".



The figure above shows the structure of one of the (identical) command tables. Up to eight commands may be defined by specifying a message ID (where to send the message to) and a parameter value. The value is not limited to be a switch position, also other choice parameters or even numerical values may be set. Hence you may change device operating modes, transmit levels or other parameters based on the single ON/OFF command directed to the Switch-Cross-Link device.

Additionally a condition may be specified for each of the commands. A command only is sent, if the actual value of the parameter specified by the condition message ID matches ("EQ", equal) or matches not ("NE", not equal) the given target value. Using conditionally commands, the effect of the primary ON/OFF command can be modified with respect to other settings like signal routings or fault conditions.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The commands to be sent if the device receives an "ON" command.
- --- The commands to be sent if the device receives an "OFF" command.
- --- The device info page.
- --- The maintenance page.

Setting up a Switch-Cross-Link logical device

The first two device window pages contain the command tables used by the device. The table below explains the parameters defined in these tables.

parameter	***description***

parameter	*description*
message id	You specify the destination of the command with the fields in this column. A message identifier is made of the name of the device followed by a dot and the name of the parameter which shall be sent. A complete list of parameter names for all device types is found later in this document. Leave this field empty if no command shall be sent.
value	In this column the value to be sent for each command is specified.
conditional parameter	You specify the message id of the conditional parameter in this column. Leave this field empty if the command shall be sent unconditionally.
comparison	If "EQ" is selected here, the command in this row only is sent if the parameter value matches the target. The selection of "NE" reverses this.
target value	The actual value of the conditional parameter gets compared against this target value to determine if the command on this row shall be sent. Be aware that this is a textual comparison, FLOAT parameters may fail to match the target value due to rounding errors.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
on	CHOICE		EnumRange (OFF ON)
out.on.0.id	TEXT	SAVE	StringRange
out.on.0.value	TEXT	SAVE	StringRange
out.on.0.cond	TEXT	SAVE	StringRange
out.on.0.target	TEXT	SAVE	StringRange
out.on.0.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.1.id	TEXT	SAVE	StringRange
out.on.1.value	TEXT	SAVE	StringRange
out.on.1.cond	TEXT	SAVE	StringRange
out.on.1.target	TEXT	SAVE	StringRange

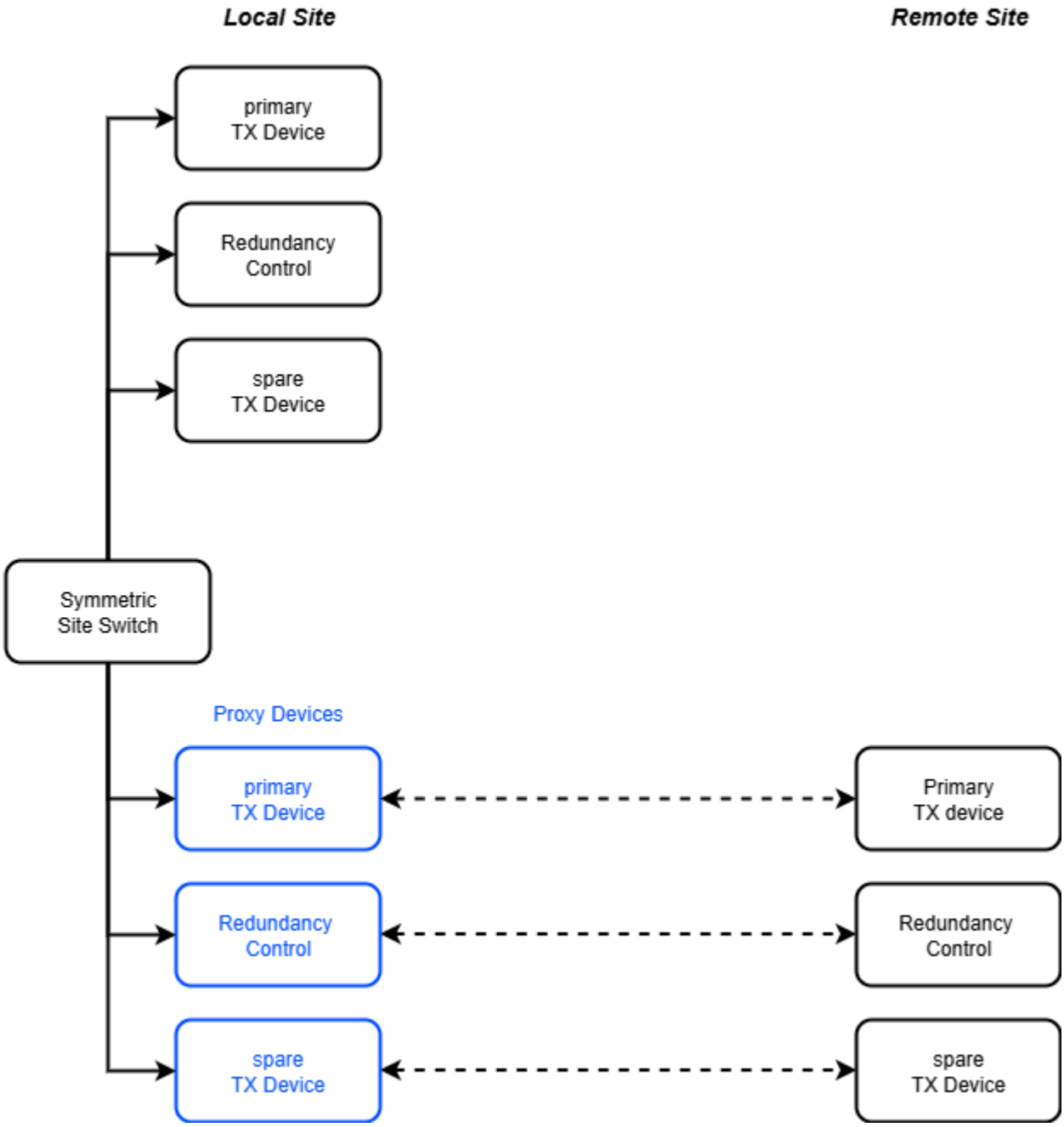
name	*type*	*flags*	*range*
out.on.1.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.2.id	TEXT	SAVE	StringRange
out.on.2.value	TEXT	SAVE	StringRange
out.on.2.cond	TEXT	SAVE	StringRange
out.on.2.target	TEXT	SAVE	StringRange
out.on.2.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.3.id	TEXT	SAVE	StringRange
out.on.3.value	TEXT	SAVE	StringRange
out.on.3.cond	TEXT	SAVE	StringRange
out.on.3.target	TEXT	SAVE	StringRange
out.on.3.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.4.id	TEXT	SAVE	StringRange
out.on.4.value	TEXT	SAVE	StringRange
out.on.4.cond	TEXT	SAVE	StringRange
out.on.4.target	TEXT	SAVE	StringRange
out.on.4.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.5.id	TEXT	SAVE	StringRange
out.on.5.value	TEXT	SAVE	StringRange
out.on.5.cond	TEXT	SAVE	StringRange
out.on.5.target	TEXT	SAVE	StringRange
out.on.5.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.6.id	TEXT	SAVE	StringRange
out.on.6.value	TEXT	SAVE	StringRange
out.on.6.cond	TEXT	SAVE	StringRange
out.on.6.target	TEXT	SAVE	StringRange
out.on.6.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.on.7.id	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
out.on.7.value	TEXT	SAVE	StringRange
out.on.7.cond	TEXT	SAVE	StringRange
out.on.7.target	TEXT	SAVE	StringRange
out.on.7.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.0.id	TEXT	SAVE	StringRange
out.off.0.value	TEXT	SAVE	StringRange
out.off.0.cond	TEXT	SAVE	StringRange
out.off.0.target	TEXT	SAVE	StringRange
out.off.0.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.1.id	TEXT	SAVE	StringRange
out.off.1.value	TEXT	SAVE	StringRange
out.off.1.cond	TEXT	SAVE	StringRange
out.off.1.target	TEXT	SAVE	StringRange
out.off.1.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.2.id	TEXT	SAVE	StringRange
out.off.2.value	TEXT	SAVE	StringRange
out.off.2.cond	TEXT	SAVE	StringRange
out.off.2.target	TEXT	SAVE	StringRange
out.off.2.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.3.id	TEXT	SAVE	StringRange
out.off.3.value	TEXT	SAVE	StringRange
out.off.3.cond	TEXT	SAVE	StringRange
out.off.3.target	TEXT	SAVE	StringRange
out.off.3.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.4.id	TEXT	SAVE	StringRange
out.off.4.value	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
out.off.4.cond	TEXT	SAVE	StringRange
out.off.4.target	TEXT	SAVE	StringRange
out.off.4.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.5.id	TEXT	SAVE	StringRange
out.off.5.value	TEXT	SAVE	StringRange
out.off.5.cond	TEXT	SAVE	StringRange
out.off.5.target	TEXT	SAVE	StringRange
out.off.5.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.6.id	TEXT	SAVE	StringRange
out.off.6.value	TEXT	SAVE	StringRange
out.off.6.cond	TEXT	SAVE	StringRange
out.off.6.target	TEXT	SAVE	StringRange
out.off.6.inv	CHOICE	SAVE	EnumRange (EQ NE)
out.off.7.id	TEXT	SAVE	StringRange
out.off.7.value	TEXT	SAVE	StringRange
out.off.7.cond	TEXT	SAVE	StringRange
out.off.7.target	TEXT	SAVE	StringRange
out.off.7.inv	CHOICE	SAVE	EnumRange (EQ NE)

9.647 Symmetric-Site-Switch

The Symmetric-Site-Switch logical device controls the equipment of one transmit chain for a site diversity application between two uplink ground stations. As the driver name implies, the device works completely symmetrically, there is no preference for a primary or a backup site.



As shown in the figure above, the Symmetric-Site-Switch logical device is operated at one of the two sites. There it controls the local RX device and the TX redundancy control device if there is any configured.

The devices of the second (remote) site are mapped asProxy devices, providing control for these devices at the other site.

Site switch procedure

Switching the uplink from the one site to site to the other is a simple sequence of actions and checks. The procedure is exactly the same for switching from one site to the other and back. In

the description below the site which transmits before the switchover is called 'source', the site which wil be transmitting after tge switchover is called 'destination'

1. Check if the destination TX equipment states a fault. Abort the the switchover in this case.
2. Switch off the protection for the TX equipment at the source site. Every type of protection (1:1, N:1, N:M) uses its own method to sitch off protection.
3. Check if the protection is actually off at the source site. Abort with an error if this fails
4. Switch off the carrier at the source site. For this first evaluate which is the tx equipment actually used for the carrier.
5. After a configurable wait time check if the carrier at the source site has been switched off. Repeatedly check this up to 10 seconds if the carrier does not respond. After this abort with an error.
6. Switch on the carrier at the destination site. For this first evaluate which is the tx equipment actually used for the carrier. protection switch if a N-TO-1 protection is configured
7. After a configurable wait time check if the carrier at the destination site has been switched on. Repeatedly check this up to 10 seconds if the carrier does not respond. After this abort with an error.
8. Restore the destination site protection on/off setting from the value which has beed saved just before the carrier had been switched away from this site the last time.
9. Check if the protection is actually set to the commanded state at the destination site. Abort with an error if this fails
10. Successfully switched to the destination site.

Switching back to the original site works the same way the other way round.

Redundancy configuration

The Symmetric-Site-Switch supports four different modes of redundancy for the transmit devices. These modes are configured individually for both sites, this means you can have two 1:1 redundant HPAs at the one site and at the other site HPAs which are part of a N:1 redundancy. The redundancy modes are:

- **NONE:** There is no redundancy at all for this site. The device only controls the primary TX device at this site and skips all steps regarding the redundancy.
- **1-TO-1:** The TX devices at this site are in a symmetric 1:1 redundancy. The site switch device always controls both TX devices at the same time, their carriers are simultaneously switched on/off, the carrier is assumed to be off if both devices report RF OFF.
- **N-TO-1:** The TX devices at this site are in a N:1 redundancy. The site switch device uses the redundant device if the protection controller has switched this in place of the configured primary device.
- **POOL**: The TX devices are controlled in a N:M redundancy by individualModem-Protection-Controller devices and a Protection-Switch-N-To-M. This works the same way for configurations which get the spare device from a pool of spares and configurations which use a 1:1 reservation of a spare device.

The protection controller mode for a site also has some impact how the Symmetric-Site-Switch expects some of its configuration parameters:

- **NONE:** txDeviceA/B contains the name of the TX device, txRedDeviceA/B and protectDeviceA/B are empty.
- **1-TO-1:** txDeviceA/B contains the name of one of the TX devices, txRedDeviceA/B the name of the other device in the 1:1 redundancy. protectDeviceA/B has to be set to the name of the 1:1 protection controller.
- **N-TO-1:** txDeviceA/B contains the name of the primary TX device, txRedDeviceA/B the name of the spare device in the N:1 redundancy. protectDeviceA/B has to set to the name of the N:1 protection controller and chainNoA/B must be set to the chain number of the primary TY device in the redundancy configuration.
- **POOL**: With the POOL based N:M redundancy, txDeviceA/B does not contain the name of a TX device but the name of the Modem-Protection-Controller managing the TX device. protectDeviceA/B has to be set to the name of the N:M protection controller, txRedDeviceA/B and chainNoA/B may be left empty.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands actual uplink location.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the device with the devices that will perform the carrier on/off settings.

parameter	***description***
abortOn	The Symmetric-Site-Switch aborts a switchover procedure if one of the transmit devices reports a fault. This setting selects if this shall happen in case of a FAULT or also if the device reports a WARNING.
protectModeA/B	Selects the type of protection switch used with the TX equipment at the particular site. May be one of NONE, 1-TO-1, N-TO-1 or POOL. Should be set to the value describing the protection mode of the TX equipment.
protectDeviceA/B	The name of the protection controller device used at the particular site. Must be set if the protectMode is set to another value than NONE

parameter	*description*
txDeviceA/B	The device name of the transmit device (typically an HPA) which produces the carrier at the particular site. With protectMode POOL this is the Modem-Protection-Controller device managing the the redundancy to this carrier
txRedDeviceA/B	The name of the backup TX device at this particular site. For a symmetrical 1:1 redundancy this is the name of the second HPA (see redundancy configuration above). Not requires if protectMode is NONE or POOL
chainNoA/B	Must be set with N:1 redundancy configuration to the chain number of the txDevice in the protection device (1..12). Not used with the other protection mode options.
nameA/B	A short (one word) name for the this site.
descriptionA/B	A descriptive name for the site. This appears in event log entries and in status messages of the device, hence it should not be to long (max 15 characters are recommended)
txChgReadDelayA/B	This parameter sets the delay between an RF ON/OFF command sent to the TX device and the first read back which is done to check if the carrier is really at the commanded state. Some devices echo the RF ON/OFF state without switching the carrier and report the true state two seconds later because the command could not be executed as expected. Increasing the delay can help to suppress the consequences of this effect.
comment	The text is presented in the device's GUI as a description of the carrier which is switched between the sites.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.protectModeA	TEXT	R/O	StringRange R/O
info.protectModeB	TEXT	R/O	StringRange R/O
info.hasBeamOnA	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.hasBeamOnB	TEXT	R/O	StringRange R/O
config.abortOn	CHOICE	SAVE SETUP	EnumRange (WARNING FAULT)
config.protectModeA	CHOICE	SAVE SETUP	EnumRange (NONE 1-TO-1 N-TO-1 POOL)
config.protectModeB	CHOICE	SAVE SETUP	EnumRange (NONE 1-TO-1 N-TO-1 POOL)
config.protectDeviceA	TEXT	SAVE SETUP	StringRange
config.protectDeviceB	TEXT	SAVE SETUP	StringRange
config.txDeviceA	TEXT	SAVE SETUP	StringRange
config.txDeviceB	TEXT	SAVE SETUP	StringRange
config.txRedDeviceA	TEXT	SAVE SETUP	StringRange
config.txRedDeviceB	TEXT	SAVE SETUP	StringRange
config.txChainNoA	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.txChainNoB	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.nameA	TEXT	SAVE SETUP	StringRange
config.nameB	TEXT	SAVE SETUP	StringRange
config.descriptionA	TEXT	SAVE SETUP	StringRange
config.descriptionB	TEXT	SAVE SETUP	StringRange
config.comment	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
gotoA	TEXT		StringRange
gotoB	TEXT		StringRange
reset	TEXT		StringRange
progressReport	TEXT	R/O	StringRange R/O
progressIcon	TEXT	R/O	StringRange R/O
location	TEXT	R/O	StringRange R/O
savedState	INTEGER	SAVE	IntegerRange (0 .. 0)
savedProtectionA	TEXT	SAVE	StringRange
savedProtectionB	TEXT	SAVE	StringRange

9.648 Symmetricom-4036B

This page describes the device driver and the device window for the Symmetricom DA4036B distribution amplifier.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	not required

Communication protocol / Interface type

This device driver requires the Terminal-CRLF-CRLF protocol. The unit will be connected via Ethernet. Default TCP/IP port is 10001.

Installation

Before you can operate the distribution amplifier with the satnms software, you must prepare the unit by setting a fixed IP address.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.input	CHOICE	R/O	EnumRange R/O (NONE 1 MHz 5 MHz 10 MHz)
internal.s	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t	TEXT	R/O	StringRange R/O
internal.i	CHOICE	R/O	EnumRange R/O (NONE 1 MHz 5 MHz 10 MHz)
faults.01	ALARM	R/O	AlarmFlagRange R/O (No Input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 7)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 15)

9.649 Symmetricom-4059B

This page describes the device driver and the device window for the Symmetricom DA4059B distribution amplifier.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	not required

Communication protocol / Interface type

This device driver requires the Terminal-CRLF-CRLF protocol. The unit will be connected via Ethernet. Default TCP/IP port is 10001.

Installation

Before you can operate the distribution amplifier with the satnms software, you must prepare the unit by setting a fixed IP address.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.input	CHOICE	R/O	EnumRange R/O (NONE 1 kHz 10 kHz Other)
internal.s	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t	TEXT	R/O	StringRange R/O
internal.i	CHOICE	R/O	EnumRange R/O (NONE 1 kHz 10 kHz Other)
faults.01	ALARM	R/O	AlarmFlagRange R/O (No Input)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Output 1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 3)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 4)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 5)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 6)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 7)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output 8)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 9)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 13)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 14)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 15)

9.650 Symmetricom-9611B

This page describes the device driver and the device window for the Symmetricom 2x12 Distribution Unit Type 9611B

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Switch parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The Symmetricom device needs some configuration parameters as well as the correct settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *115200,N,8,1* .
impedance	Define if 50Ohm or 1kOhm impedance should be used.

Remote Control

- The serial parameters are: RS232,115.2k,8,None,1,None.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.impedance	CHOICE	SETUP	EnumRange (50 Ohm 1000 Ohm)
info.model	TEXT	R/O	StringRange R/O
info.versionMicro	TEXT	R/O	StringRange R/O
info.versionFpga	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange (1Hz to < 10Hz 10Hz to < 100Hz 100Hz to < 1kHz 1kHz to < 10kHz 10kHz to < 100kHz 100kHz and above Digital IRIG A Digital IRIG B Digital IRIG D Digital IRIG E Digital IRIG G Digital IRIG H AM IRIG - 100Hz carrier AM IRIG - 1kHz carrier AM IRIG - 10kHz carrier)
position	CHOICE		EnumRange (A B)
protection	CHOICE		EnumRange (AUTO USER)
internal.faultList	TEXT	R/O	StringRange R/O
int.fault1	TEXT	R/O	StringRange R/O
int.fault2	TEXT	R/O	StringRange R/O
internal.position	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Output 01)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output 02)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 03)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 04)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 05)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 06)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output 07)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output 08)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output 09)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output 10)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 11)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 12)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.651 Symmetricom-S650

This page describes the device driver and the device window for Microsemi*Symmetricom-S650* Synchronization Server via SNMP. The driver is Read Only.

Please note: SNMP is disabled by factory default.

It was tested with:

- SW 5.1.2

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sensor readings.
- --- The faults page.
- --- The device info page.

- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	description
address	The SNMP address specification for the device (details below).
sensors	Defines the number pf installed sensor to be monitored.
sensor.1..4.loTemp	Defines the low temperature threshold in °C. If this value is undercut a low temperature alarm is raised
sensor.1..4.hiTemp	Defines the high temperature threshold in °C. If this value is exceeded a high temperature alarm is raised
sensor.1..4.loHumidity	Defines the low humidity threshold in %. If this value is undercut a low humidity alarm is raised
sensor.1..4.hiHumidity	Defines the high humidity threshold in %. If this value is exceeded a high humidity alarm is raised
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.

parameter	description
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. The device uses `read` as a default for read and `write` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.slotA	TEXT	R/O	StringRange R/O
info.slotB	TEXT	R/O	StringRange R/O
info.oscillator	TEXT	R/O	StringRange R/O
info.psu	TEXT	R/O	StringRange R/O
info.ntpVersion	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
ntp.leap	TEXT	R/O	StringRange R/O
ntp.stratum	INTEGER	R/O	IntegerRange R/O (0 .. 255)

name	*type*	*flags*	*range*
ntp.clkPrecision	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.dispersion	TEXT	R/O	StringRange R/O
ntp.delay	TEXT	R/O	StringRange R/O
ntp.refId	TEXT	R/O	StringRange R/O
ntp.refTime	TEXT	R/O	StringRange R/O
ntp.peer	TEXT	R/O	StringRange R/O
ntp.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.sysMode	TEXT	R/O	StringRange R/O
clock.state	TEXT	R/O	StringRange R/O
clock.reference	TEXT	R/O	StringRange R/O
clock.time	TEXT	R/O	StringRange R/O
gnss.position	TEXT	R/O	StringRange R/O
gnss.receiver	TEXT	R/O	StringRange R/O
gnss.noSat	INTEGER	R/O	IntegerRange R/O (0 .. 0)
gnss.satlist	TEXT	R/O	StringRange R/O
gnss.state	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (NTP Sys Leap)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Clock State)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Clock State Warning)
faults.04	ALARM	R/O	AlarmFlagRange R/O (GNSS Receiver)
faults.05	ALARM	R/O	AlarmFlagRange R/O (GNSS No Sat)

9.652 Symmetricom-XLI

This page describes the device driver and the device window for the Symmetricom XLI Time and frequency system.

The driver was tested in 1:1 configuration with the following SW Releases:

- SW: 192-8001

Please note that SNMP is disabled by factory default. Therefore change snmp.conf via WebUI.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device switch parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- timeThreshold --- Defines the limit in nanoseconds, for the XLi clocks accumulated time error
- timeoutDelay --- Defines the delay time for exceeding the timeThreshold before generating a time error alarm
- poweronAlarm --- Defines the time after a power on cycle, for how long F73 alarms are suppressed.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.timeThreshold	INTEGER	SETUP	IntegerRange (0 .. 99999)
config.timeoutDelay	INTEGER	SETUP	IntegerRange (0 .. 86400)
config.powerOnAlarm	INTEGER	SETUP	IntegerRange (0 .. 86400)
internal.poll	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.precision	INTEGER	R/O	IntegerRange R/O (0 .. 0)
source	CHOICE	R/O	EnumRange R/O (IRIG-A IRIG-B IRIG-G NASA-36 PRIMARY SECONDARY AUXREF NONE)
state	TEXT	R/O	StringRange R/O
ntp.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.packetsRxErr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.packetsRxAuth	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.minAccuracy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.estAccuracy	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.leap	CHOICE	R/O	EnumRange R/O (NO LEAP SEC INSERTED SEC DELETED LOSS OF TIMING SYNC)

name	*type*	*flags*	*range*
ntp.hostMode	CHOICE	R/O	EnumRange R/O (RESERVED ACTIVE PASSIVE CLIENT SERVER BROADCAST)
ntp.stratum	INTEGER	R/O	IntegerRange R/O (0 .. 255)
ntp.poll	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ntp.clkPrecision	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ntp.delay	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ntp.dispersion	FLOAT	R/O	DoubleRange R/O (0 .. 0)
ntp.clkId	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
faults.02	ALARM	R/O	AlarmFlagRange R/O (PLL lock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (PLL lock Latched)
faults.03	ALARM	R/O	AlarmFlagRange R/O (LPN PLL lock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (LPN PLL lock Latched)
faults.04	ALARM	R/O	AlarmFlagRange R/O (GPS primary RX)
faults.18	ALARM	R/O	AlarmFlagRange R/O (GPS pri. RX Latched)
faults.05	ALARM	R/O	AlarmFlagRange R/O (GPS secondary RX)
faults.19	ALARM	R/O	AlarmFlagRange R/O (GPS sec. RX Latched)
faults.06	ALARM	R/O	AlarmFlagRange R/O (IRIG)
faults.20	ALARM	R/O	AlarmFlagRange R/O (IRIG Latched)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Aux reference)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Aux ref. Latched)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Primary power)

name	*type*	*flags*	*range*
faults.22	ALARM	R/O	AlarmFlagRange R/O (Prim. power Latched)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Secondary power)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Sec. power Latched)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Rubidium Osc)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Rb Osc Latched)
faults.11	ALARM	R/O	AlarmFlagRange R/O (DAC)
faults.25	ALARM	R/O	AlarmFlagRange R/O (DAC Latched)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Time lock since on)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Time lock since on Latched)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Time)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Time Latched)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Timeout)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Timeout Latched)
faults.15	ALARM	R/O	AlarmFlagRange R/O (NTP)
faults.29	ALARM	R/O	AlarmFlagRange R/O (NTP Latched)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Clock)

9.653 Sync-Switch

The Sync-Switch logical device couples two switches located at two different sites (i.e. controlled by two different M&C/VLC systems) to one single switch. A possible application for this function is a manually controlled site diversity switch. With one single parameter, the local transmission is switched on while at the remote site the transmission gets switch off and vice versa.

Sync-Switch devices may be used symmetrically which means that at both sites a Sync-Switch device is configured which controls both switches. It also is possible to configure the Sync-Switch only at one of the two sites.

The Sync-Switch device monitors the 'sanity' of both switches, the availability of the remote network connection and the consistency of the switch settings. It sets the applicable fault flag if it detects an error condition.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control page for the device. The operational parameters of the Sync-Switch device are described below.
- --- The fault page.
  - **Remote connection** : The TCP/IP connection to the remote site is broken.
  - **Switch combination** : The switches are not in a consistent position.
  - **Local switch** : Fault of the local switch.
  - **Remote switch** : Fault of the remote switch.
- --- The device info page.
- --- The maintenance page. The configuration parameters described later are set at this page.

Operational parameters

The operational parameters of the Sync-Switch device are set at the main control page. The table below explains each parameter in particular.

parameter	***description***
position	Like for an ordinary 'Switch' device, 'position' is the only operational parameter. Changing this value, actuates the local and the remote switch as well. Which choices this parameter offers may be configured with the parameters below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Sync-Switch device with the device id shall monitor/control.

parameter	***description***
choices	With this configuration parameter you define which choices appear in the drop down box for the 'position' parameter. You may want to set the names of the two sites here, for example. In the choices field, two words separated by a single comma must appear. There should be no spaces around the comma as the program would treat them as part of the choices.

parameter	*description*
localSwitchName	This is the device name of the local switch device to be controlled.
localSwitchChoices	In this field, two position values for the local switch, separated by a comma, must be set. The position values mut appear in that order that matches the values in 'choices'. The program will send the first position value to the switch if the operator selects the first choice value of 'position'.
remoteSwitchName	This is the device name of the switch device at the remote site to be controlled.
remoteSwitchChoices	In this field, two position values for the remote switch, separated by a comma, must be set. The position values mut appear in that order that matches the values in 'choices'. The program will send the first position value to the switch if the operator selects the first choice value of 'position'.
remoteIpAddress	Specify the IP address of the remote M&C computer with this parameter, either in dotted quad notation or as a computer name the local machine is able to resolve.

Remarks

- The Sync-Switch devices require a *permanent* network connection between the two sites. They are sensing the presence of this connection with a cyclic "ping" and will report a fault if the network link is down. If you are using an "on demand" dial-up link between the M&C computers, the ping probably will keep the link alive all the time.
- The 'Switch combination' fault responds very fast. The fault will be activated for a short time with each switch actuation. This is because the local and the remote switch don't change their positions at the same time. Set the fault delay parameter of this fault flag to value of one or two seconds to avoid this behavior.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.choices	TEXT	SAVE SETUP	StringRange
config.localSwitchName	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.localSwitchChoices	TEXT	SAVE SETUP	StringRange
config.remoteSwitchName	TEXT	SAVE SETUP	StringRange
config.remoteSwitchChoices	TEXT	SAVE SETUP	StringRange
config.remoteIpAddress	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote connection)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Switch combination)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Local switch)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Remote switch)

9.654 Sysinfo

The Sysinfo logical device displays parameters of the server machine. This includes disk usage, cpu load, memory and sensor information from the mainboard if available on the specific hardware platform.

It is possible to configure alarm thresholds to create alarm events if the disk capacity goes low, if the 15min average cpu load is to high, if the available memory is to low or if a temperature is to high.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows cpu load and disk usage etc.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

parameter	*description*
diskThreshold	The threshold for the available disk capacity
loadThreshold	The threshold for the 15min average cpu load
memThreshold	The threshold for RAM
core0Threshold	The threshold for CPU temperature core0 if available
core1Threshold	The threshold for CPU temperature core1 if available
core1Threshold	The threshold for mainboard temperature 1 if available
core1Threshold	The threshold for mainboard temperature 2 if available
core1Threshold	The threshold for mainboard temperature 3 if available

Fault flags

The device provides the following fault flags:

- 'low disk capacity': the available space on the internal disk is lower than the configured threshold
- 'high average load': the average cpu load of the last 15min was higher than the configured threshold
- 'low memory': the available memory (RAM) is lower than the configured threshold
- 'high .. temperature': the temperature is higher than the configured threshold

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.hw	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.hostname	TEXT	R/O	StringRange R/O
info.cores	TEXT	R/O	StringRange R/O
info.devLicUsed	TEXT	R/O	StringRange R/O
config.execHost	CHOICE	SAVE SETUP	EnumRange (LOCAL REMOTE)
config.diskName	TEXT	SAVE SETUP	StringRange
config.diskThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loadThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.memThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.core0Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.core1Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp1Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp2Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.temp3Threshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
disk.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
disk.size	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.available	INTEGER	R/O	IntegerRange R/O (0 .. 0)
disk.usage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.total	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.used	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mem.free	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.uptime	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
stat.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
stat.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
temperature.core0	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.core1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temperature.3	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fan.speed.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.speed.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
voltage.in0	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in3	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in4	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
voltage.in5	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in6	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.in7	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
voltage.vbat	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ps.state.1	TEXT	R/O	StringRange R/O
ps.state.2	TEXT	R/O	StringRange R/O
internal.ps1	TEXT	R/O	StringRange R/O
internal.ps2	TEXT	R/O	StringRange R/O
mail.count	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mail.subjects	TEXT	R/O	StringRange R/O
mnc.redundancy	CHOICE	R/O	EnumRange R/O (ERROR NONE MAIN BACKUP)
mnc.state	CHOICE	R/O	EnumRange R/O (ERROR ACTIVE STANDBY)
mnc.remote	TEXT	R/O	StringRange R/O
mnc.mainAddress	TEXT	R/O	StringRange R/O
mnc.backupAddress	TEXT	R/O	StringRange R/O
backend.redundancy	CHOICE	R/O	EnumRange R/O (ERROR NONE MAIN BACKUP)
backend.state	CHOICE	R/O	EnumRange R/O (ERROR ACTIVE STANDBY)
backend.remote	TEXT	R/O	StringRange R/O
backend.mainAddress	TEXT	R/O	StringRange R/O
backend.backupAddress	TEXT	R/O	StringRange R/O
backend.running	CHOICE	R/O	EnumRange R/O (YES NO)
internal.version	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.os	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.path	TEXT	R/O	StringRange R/O

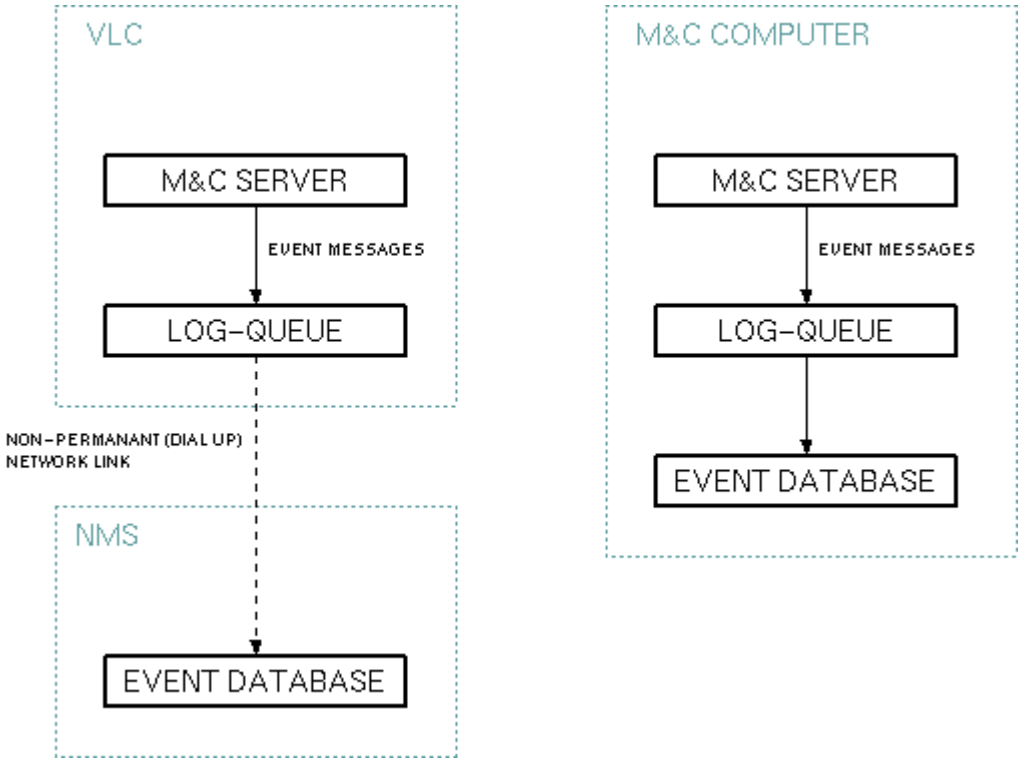
name	*type*	*flags*	*range*
internal.loadAvg1min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg5min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.loadAvg15min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (low disk capacity)
faults.02	ALARM	R/O	AlarmFlagRange R/O (high average load)
faults.03	ALARM	R/O	AlarmFlagRange R/O (high core 0 temp)
faults.04	ALARM	R/O	AlarmFlagRange R/O (high core 1 temp)
faults.05	ALARM	R/O	AlarmFlagRange R/O (high board 1 temp)
faults.06	ALARM	R/O	AlarmFlagRange R/O (high board 2 temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (high board 3 temp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (low memory)
faults.09	ALARM	R/O	AlarmFlagRange R/O (power supply 1)
faults.10	ALARM	R/O	AlarmFlagRange R/O (power supply 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (ps 1 not installed)
faults.12	ALARM	R/O	AlarmFlagRange R/O (ps 2 not installed)
faults.13	ALARM	R/O	AlarmFlagRange R/O (new mail)
faults.21	ALARM	R/O	AlarmFlagRange R/O (fan 1)
faults.22	ALARM	R/O	AlarmFlagRange R/O (fan 2)
faults.23	ALARM	R/O	AlarmFlagRange R/O (fan 3)
faults.24	ALARM	R/O	AlarmFlagRange R/O (fan 4)

name	*type*	*flags*	*range*
faults.25	ALARM	R/O	AlarmFlagRange R/O (fan 5)
faults.26	ALARM	R/O	AlarmFlagRange R/O (mnc switched)
faults.27	ALARM	R/O	AlarmFlagRange R/O (backup mnc)
faults.28	ALARM	R/O	AlarmFlagRange R/O (backend switches)
faults.29	ALARM	R/O	AlarmFlagRange R/O (backup backend)
faults.30	ALARM	R/O	AlarmFlagRange R/O (backend running)

9.655 System

The System device with the name SYSTEM is present on every M&C/VLC server. It does not appear in the "vlc.setup" file, the server creates it automatically as a gateway to some of it's internal services. You mainly use the system device to tell the M&C/VLC server on which way and on which conditions the server shall send event messages to the event log.

To understand the parameters provided by the System device, it is necessary to know that the M&C/VLC server routes all faults and other messages through the so called "log-queue".



With a NMS/VLC network control system, all event messages are stored at a central place at the NMS. As the VLC usually has no permanent data link to the NMS, the log-queue holds the messages until a network link to the NMS is available and the queue gets flushed to the central event database.

The VLC is capable to dial to the NMS by itself. To limit the costs of the dial up connections, time delays may be specified to prevent the VLC from dialing with each message that occurs. If the log-queue gets flushed (which implies a dial-up from the VLC to the NMS) depends on the following. The queue gets flushed if:

- it contains at least one message of priority "ALARM", regardless of the age of the message.
- it contains at least one message of priority "FAULT" which is older than the parameter "fault delay".
- it contains at least one message of priority "INFO" which is older than the parameter "informational delay".
- the queue is full.
- a dial-up connection setup from NMS becomes on-line.

With a M&C system, the M&C computer itself provides the event database in most cases. The M&C server still routes the event messages through the log-queue, the timing parameter described above still apply, but they are set in a way that all messages go straight through into the event database.

Clock synchronisation

Another task of the SYSTEM device is to control the synchronization of the VLC's clock to the NMS time. This synchronization is essential for the link management as scheduled links are planned on the base of the NMS' clock but executed following the VLC's clock. The synchronization works as follows:

Each time, the NMS connects to a VLC, it sends the actual time to the VLC. The SYSTEM device compares the local time to the time received from the NMS and -- depending on the difference -- tunes the VLC's clock a little bit faster or slower. With VLCs being accessed at least once a month, this keeps the VLC's clock well in tune with the NMS without producing any leaps in the local time. This clock synchronization is made once when a VLC becomes 'on-line'.

If the clocks of NMS and VLC differ by more than 30 seconds, the SYSTEM device treas this as an error. It states a FAULT (it's icon at the user interface becomes red) and writes a message to the event log with a message text like:

Clock differs by 1523 secs from NMS time. Please reboot.

There may be several reasons for this fault:

- You have set the NMS clock manually, now each VLC states this fault until all VLCs are in tune again.
- The VLC was not accessed for a long time. A time drift of 15 secs / month is quite usual.
- The same applies if the VLC has been on-line for a very long time without interruption. As

the clock synchronization only happens when the VLC *becomes* on-line, the clock will drift out of tune in this case.

When the SYSTEM device has recognized this fault, the VLC already is about to adjust the clock to the right time. It however cannot set the correct time immediately because the time leap could mess up the software's timeout recognition. The VLC smoothly tunes the clock with a maximum rate of 10 minutes per day. Hence, with relative small time differences you might ignore the fault, for larger differences a re-boot of the VLC will speed up this procedure, the correct time is set when the VLC starts (see section 'SYSTEM Device fault' below for details).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The parameters at this page define on which way event messages shall be sent to the event database.
    - NMSnetworkaddress --- Defines the host name / IP address of the NMS / event database server.
        - If you are configuring a VLC and this unit is connected to the NMS via Ethernet or a permanent SCC link, enter the IP address of the NMS.
        - For a VLC exclusively using dial up connections to the NMS enter the word "offline".
        - For M&C computers enter "localhost" or "127.0.0.1".
    - dial command 1 dial command 2 dial command 3 --- For a VLC you can specify up to three dial commands (AT.. command sequences) the VLC shall use to connect to the NMS. The VLC always tries dial command 1 first, if this fails it tries the second one an so on. You must enter the complete modem AT command, not only the phone number. This lets you prepend some additional commands like "X3" or some switches in front of the "D" command.With a M&C system or a VLC which communicates to NMS via Ethernet, let all three dial commands empty. This prevents the machine from doing any dial attempts.
- --- Contains the size and timing parameter of the log queue.
    - size --- The number of event messages the queue can hold. If the queue is full, the last (youngest) message in the queue gets replaced by a overflow message and all further messages are discarded.
    - fault delay --- The time delay for FAULT messages. With a M&C system, you should set this value to 0.
    - informational --- The time delay for INFO messages. With a M&C system, you should set this value to 0.
    - redial delay --- Defines the pause between two dialing attempts a VLC makes. This should be set to about 5 minutes, some modems decline more frequent dialing commands to the same phone number.

- --- Contains the macro backup function which permits to store all parameter settings known to the M&C/VLC into a macro definition for later retrieval.
- --- The device info page. You see the software revision number/date of the M&C/VLC server at this place

SYSTEM Device Fault

If the SYSTEM device states a fault, this indicates that the clocks of NMS and VLC are not in tune. The VLC will correct the clock difference automatically after some time, but actually any scheduled links from this VLC will start or stop at the wrong time.

If the time difference is relative small you may ignore the fault. The time difference is stated in the event log, search for the VLC ID and the text 'Clock differs'. Clicking the 'RESET' button on 'Maintenance' page of the device window will reset the fault until the VLC goes on-line the next time.

If the time difference is larger or if you need the clock to be set precisely at once, you may re-boot the VLC. This means to restart the VLC's operating system, not only to restart the VLC application. The procedure for this is as follows:

1. Establish an on-line connection to the VLC.
2. Open a telnet window to the VLC (Depending on your network settings, this might require that you are working directly at the NMS server).
3. Log in as user 'satnms' (password 'satnms').
4. Enter 'su', enter the password 'root' when prompted.
5. Enter 'reboot'.

The connection to the VLC will break after this, about 3 minutes later the VLC will be accessible again with the clock set properly.

Remarks

- When setting up a M&C system, ensure that "localhost" is set as the NMS network address and both delays for FAULT and INFO messages are set to 0. You will see the event messages delayed or event not at all in the event log if you leave these parameters unconfigured.
- VLC may be configured to talk to the NMS using a permanent network link and a dial up connection as a fall back. You might use this constellation with an SCC connection. If the satellite link fails, the VLC tries to use the terrestrial phone line to send it's fault messages.
- Setting the "fault delay" and "informational delay" parameters for a VLC which uses a dial-up connection to the NMS always is a compromise. If you are small values, you get informed earlier about malfunctions in the VLC, but there will be higher connection costs.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O

9.656 T21-9261-Decoder

This page describes the device driver and the device window for the Vitec T21 IP/SRT decoder, controlled via SNMP.

The driver handles versions with one channel and up to four single output transport streams. For dual encoding use two device instances and select channel via devic setup.

The driver was tested with sw release**Major 2.2, Minor 0.74.0**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the general input and program selection.
- --- Contains RTP/UDP/SRT input parameters
- --- Contains the parameters concerning the video decoding.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the T21.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write

community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.versionMinor	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.key	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.coding	TEXT	R/O	StringRange R/O
internal.string	TEXT	R/O	StringRange R/O
internal.string2	TEXT	R/O	StringRange R/O
internal.string3	TEXT	R/O	StringRange R/O
internal.mode	TEXT	R/O	StringRange R/O
internal.apid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.redId	TEXT	R/O	StringRange R/O
internal.audio	TEXT	R/O	StringRange R/O
internal.level.l	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level.r	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
restart	TEXT	NOPRESET	StringRange
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (00000 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
input.redundancy	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
input.active	CHOICE	R/O	EnumRange R/O (MAIN BACKUP)
input.1.type	CHOICE		EnumRange (N/A SRT IP RTMP ZIXI)
input.2.type	CHOICE		EnumRange (N/A SRT IP RTMP ZIXI)
input.buffer	INTEGER		IntegerRange (0 .. 0)
input.state	CHOICE	R/O	EnumRange R/O (OFF ON)
input.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
input.packetsRx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsSequ	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsDupl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsOver	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.sentNAK	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.sentACK	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input.mtu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.format	TEXT	R/O	StringRange R/O
video.frameRate	TEXT	R/O	StringRange R/O
video.codec	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8190)
video.aspectRatio	TEXT	R/O	StringRange R/O
video.interlace	CHOICE		EnumRange (PROGRESSIVE INTERLACE)
video.datarate	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
video.failMode	CHOICE		EnumRange (CLOSE FREEZE COLOR+TXT IMAGE COLOR)
video.fail	CHOICE		EnumRange (BLACK WHITE YELLOW CYAN MAGENTA RED BLUE GREEN GRAY)
output.conn	CHOICE		EnumRange (AUTO MANUAL)
output.aspectRatio	CHOICE		EnumRange (16:9 4:3)
video.arConvert	CHOICE		EnumRange (ORIGINAL FIT LETTERBOX PAN SCALE)
ipin.1.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipin.1.address	TEXT		StringRange
ipin.1.port	INTEGER		IntegerRange (0 .. 65535)
ipin.1.protocol	CHOICE		EnumRange (RTP UDP TCP)
srtin.1.address	TEXT		StringRange
srtin.1.port	INTEGER		IntegerRange (0 .. 65535)
srtin.1.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
srtin.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtin.1.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtin.1.key	TEXT		StringRange
srtin.1.stream	TEXT		StringRange
srtin.1.timeout	INTEGER		IntegerRange (0 .. 0)
srtin.1.latency	INTEGER		IntegerRange (0 .. 0)
rtmpin.1.url	TEXT		StringRange
zixiin.1.stream	TEXT		StringRange
zixiin.1.remote	TEXT		StringRange
zixiin.1.password	TEXT		StringRange
zixiin.1.address	TEXT		StringRange

name	*type*	*flags*	*range*
zixiin.1.port	INTEGER		IntegerRange (0 .. 65535)
zixiin.1.latency	INTEGER		IntegerRange (0 .. 0)
zixiin.1.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
zixiin.1.key	TEXT		StringRange
zixiin.1.fecEnable	CHOICE		EnumRange (OFF ON)
zixiin.1.fecAware	CHOICE		EnumRange (OFF ON)
zixiin.1.fecOverhead	INTEGER		IntegerRange (0 .. 0)
zixiin.1.fecBlock	INTEGER		IntegerRange (0 .. 0)
ipin.2.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipin.2.address	TEXT		StringRange
ipin.2.port	INTEGER		IntegerRange (0 .. 65535)
ipin.2.protocol	CHOICE		EnumRange (RTP UDP)
srtin.2.address	TEXT		StringRange
srtin.2.port	INTEGER		IntegerRange (0 .. 65535)
srtin.2.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
srtin.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtin.2.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtin.2.key	TEXT		StringRange
srtin.2.stream	TEXT		StringRange
srtin.2.timeout	INTEGER		IntegerRange (0 .. 0)
srtin.2.latency	INTEGER		IntegerRange (0 .. 0)
zixiin.2.stream	TEXT		StringRange
zixiin.2.remote	TEXT		StringRange
zixiin.2.password	TEXT		StringRange
zixiin.2.address	TEXT		StringRange

name	*type*	*flags*	*range*
zixiin.2.port	INTEGER		IntegerRange (0 .. 65535)
zixiin.2.latency	INTEGER		IntegerRange (0 .. 0)
zixiin.2.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
zixiin.2.key	TEXT		StringRange
zixiin.2.fecEnable	CHOICE		EnumRange (OFF ON)
zixiin.2.fecAware	CHOICE		EnumRange (OFF ON)
zixiin.2.fecOverhead	INTEGER		IntegerRange (0 .. 0)
zixiin.2.fecBlock	INTEGER		IntegerRange (0 .. 0)
rtmpin.2.url	TEXT		StringRange
output.enable	CHOICE		EnumRange (OFF ON)
output.type	CHOICE		EnumRange (N/A SRT IP ZIXI)
output.state	CHOICE	R/O	EnumRange R/O (OFF ON)
output.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
output.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipout.address	TEXT		StringRange
ipout.port	INTEGER		IntegerRange (0 .. 65535)
ipout.protocol	CHOICE		EnumRange (RTP UDP)
ipout.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.rateControl	CHOICE		EnumRange (OFF ON)
ipout.bitrate	FLOAT		DoubleRange (0.000 .. 0.000)
srtout.address	TEXT		StringRange
srtout.port	INTEGER		IntegerRange (0 .. 65535)
srtout.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)

name	*type*	*flags*	*range*
srtout.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.tos	TEXT		StringRange
srtout.mtu	INTEGER		IntegerRange (0 .. 0)
srtout.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtout.key	TEXT		StringRange
srtout.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.stream	TEXT		StringRange
srtout.timeout	INTEGER		IntegerRange (0 .. 0)
srtout.latency	INTEGER		IntegerRange (0 .. 0)
srtout.inputBw	INTEGER		IntegerRange (0 .. 0)
srtout.maxBw	INTEGER		IntegerRange (0 .. 0)
zixiout.stream	TEXT		StringRange
zixiout.remote	TEXT		StringRange
zixiout.password	TEXT		StringRange
zixiout.address	TEXT		StringRange
zixiout.port	INTEGER		IntegerRange (0 .. 65535)
zixiout.latency	INTEGER		IntegerRange (0 .. 0)
zixiout.encryption	CHOICE		EnumRange (NONE AES-128 AES-192 AES-256)
zixiout.key	TEXT		StringRange
zixiout.fecEnable	CHOICE		EnumRange (OFF ON)
zixiout.fecAware	CHOICE		EnumRange (OFF ON)
zixiout.fecOverhead	INTEGER		IntegerRange (0 .. 0)
zixiout.fecBlock	INTEGER		IntegerRange (0 .. 0)
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.1.pidSel	CHOICE		EnumRange (OFF AUTO)
audio.1.mode	CHOICE		EnumRange (STEREO-PCM PASSTHROUGH)
audio.1.delay	INTEGER		IntegerRange (0 .. 0)
audio.1.routing	CHOICE		EnumRange (AUTO LtRt LoRo USER)
audio.1.dynamic	CHOICE		EnumRange (DEFAULT LINE RF)
audio.1.dynamicRange	CHOICE		EnumRange (DEFAULT KEEP ON OFF)
audio.1.samplingRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.datarate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.pidSel	CHOICE		EnumRange (OFF AUTO)
audio.2.mode	CHOICE		EnumRange (STEREO-PCM PASSTHROUGH)
audio.2.delay	INTEGER		IntegerRange (0 .. 0)
audio.2.routing	CHOICE		EnumRange (AUTO LtRt LoRo USER)
audio.2.dynamic	CHOICE		EnumRange (DEFAULT LINE RF)
audio.2.dynamicRange	CHOICE		EnumRange (DEFAULT KEEP ON OFF)
audio.2.samplingRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.datarate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.3.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.pidSel	CHOICE		EnumRange (OFF AUTO)
audio.3.mode	CHOICE		EnumRange (STEREO-PCM PASSTHROUGH)
audio.3.delay	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
audio.3.routing	CHOICE		EnumRange (AUTO LtRt LoRo USER)
audio.3.dynamic	CHOICE		EnumRange (DEFAULT LINE RF)
audio.3.dynamicRange	CHOICE		EnumRange (DEFAULT KEEP ON OFF)
audio.3.samplingRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.datarate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.4.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.pidSel	CHOICE		EnumRange (OFF AUTO)
audio.4.mode	CHOICE		EnumRange (STEREO-PCM PASSTHROUGH)
audio.4.delay	INTEGER		IntegerRange (0 .. 0)
audio.4.routing	CHOICE		EnumRange (AUTO LtRt LoRo USER)
audio.4.dynamic	CHOICE		EnumRange (DEFAULT LINE RF)
audio.4.dynamicRange	CHOICE		EnumRange (DEFAULT KEEP ON OFF)
audio.4.samplingRate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.datarate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Transport Stream)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 4)

9.657 T21-9261-Encoder

This page describes the device driver and the device window for the Vitec T21**RTP/UDP/SRT** encoder, controlled via SNMP. The following formats are not supported:
RTMP/HLS/ZIXI/RTSP/TCP SERVER

The driver was tested with sw release**Major 2.2, Minor 0.74.0**

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the MPEG output parameter settings and the operation control.
- --- Contains RTP/UDP output parameters
- --- Contains SRT output parameters
- --- Contains the parameters concerning the video encoding.
- --- Contains the parameters concerning the audio channel 1 encoding.
- --- Contains the parameters concerning the audio channel 2 encoding.
- --- Contains the parameters concerning the audio channel 3 encoding.
- --- Contains the parameters concerning the audio channel 4 encoding.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the T21.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters

recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- channel --- Defines the selected channel for dual channel units.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.versionMinor	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.channel	CHOICE	SAVE SETUP	EnumRange (1 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.channel	CHOICE	R/O	EnumRange R/O (0 1)
internal.key	TEXT	R/O	StringRange R/O
internal.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.string	TEXT	R/O	StringRange R/O
internal.mode	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
restart	TEXT	NOPRESET	StringRange
video.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
video.source	TEXT	R/O	StringRange R/O
audio.1.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.7.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.8.inputDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.datarate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.netname	TEXT		StringRange
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)
output.shaping	CHOICE		EnumRange (ON OFF)
output.1.type	CHOICE		EnumRange (N/A SRT IP)
output.2.type	CHOICE		EnumRange (N/A SRT IP)
output.3.type	CHOICE		EnumRange (N/A SRT IP)

name	*type*	*flags*	*range*
output.4.type	CHOICE		EnumRange (N/A SRT IP)
output.1.enable	CHOICE		EnumRange (OFF ON)
output.2.enable	CHOICE		EnumRange (OFF ON)
output.3.enable	CHOICE		EnumRange (OFF ON)
output.4.enable	CHOICE		EnumRange (OFF ON)
output.1.state	CHOICE	R/O	EnumRange R/O (OFF ON)
output.1.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
output.1.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.2.state	CHOICE	R/O	EnumRange R/O (OFF ON)
output.2.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
output.2.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.3.state	CHOICE	R/O	EnumRange R/O (OFF ON)
output.3.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
output.3.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.4.state	CHOICE	R/O	EnumRange R/O (OFF ON)
output.4.datarate	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
output.4.packetCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.sdiMode	CHOICE		EnumRange (AUTO FIX)
video.sdiFormat	CHOICE		EnumRange (1080p60 1080p59.94 1080i60 1080i59.94 1080p50 1080i50 1080p30 1080p29.97 1080p25 1080p24 1080p23.976 720p60 720p59.94 720p50 720p30 720p29.97 720p25 720x576i50 720x480i60 720x480i59.94)

name	*type*	*flags*	*range*
video.format	CHOICE		EnumRange (AUTO 1920x1080 1440x1080 1280X720 720x576 720x480 640x480 320x240 1024x576 960x540 768x432 512x288)
video.frameRate	CHOICE		EnumRange (AUTO 60fps 59.94fps 50fps 30fps 29.97fps 25fps 24fps 23.976fps 20fps 15fps 10fps 5fps)
video.interlace	CHOICE		EnumRange (DEINTERLACE INTERLACE AUTO)
video.rateControl	CHOICE		EnumRange (CBR VBR)
video.codec	CHOICE		EnumRange (MPEG2 MPEG4-AVC)
video.profile	CHOICE		EnumRange (BASELINE CONSTRAINED BASELINE EXTENDED MAIN HIGH SIMPLE MAIN)
video.level	CHOICE		EnumRange (1 1B 1.1 1.2 1.3 2 2.1 2.2 3 3.1 3.2 4 4.1 4.2 LOW MAIN HIGH14 HIGH)
video.framesB	INTEGER		IntegerRange (0 .. 3)
video.framesP	INTEGER		IntegerRange (0 .. 254)
video.gop	CHOICE		EnumRange (CLOSED OPEN)
video.datarate	FLOAT		DoubleRange (0.0000 .. 0.0000)
video.pid	INTEGER		IntegerRange (0 .. 8190)
ipout.1.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.protocol	CHOICE		EnumRange (RTP UDP)
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.1.address	TEXT		StringRange
srtout.1.port	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.1.tos	TEXT		StringRange
srtout.1.mtu	INTEGER		IntegerRange (0 .. 0)
srtout.1.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtout.1.key	TEXT		StringRange
srtout.1.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.1.stream	TEXT		StringRange
srtout.1.timeout	INTEGER		IntegerRange (0 .. 0)
srtout.1.latency	INTEGER		IntegerRange (0 .. 0)
srtout.1.inputBw	INTEGER		IntegerRange (0 .. 0)
srtout.1.maxBw	INTEGER		IntegerRange (0 .. 0)
ipout.2.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 65535)
ipout.2.protocol	CHOICE		EnumRange (RTP UDP)
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.2.address	TEXT		StringRange
srtout.2.port	INTEGER		IntegerRange (0 .. 65535)
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.2.tos	TEXT		StringRange
srtout.2.mtu	INTEGER		IntegerRange (0 .. 0)
srtout.2.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)

name	*type*	*flags*	*range*
srtout.2.key	TEXT		StringRange
srtout.2.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.2.stream	TEXT		StringRange
srtout.2.timeout	INTEGER		IntegerRange (0 .. 0)
srtout.2.latency	INTEGER		IntegerRange (0 .. 0)
srtout.2.inputBw	INTEGER		IntegerRange (0 .. 0)
srtout.2.maxBw	INTEGER		IntegerRange (0 .. 0)
ipout.3.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipout.3.address	TEXT		StringRange
ipout.3.port	INTEGER		IntegerRange (0 .. 65535)
ipout.3.protocol	CHOICE		EnumRange (RTP UDP)
ipout.3.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.3.address	TEXT		StringRange
srtout.3.port	INTEGER		IntegerRange (0 .. 65535)
srtout.3.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.3.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.3.tos	TEXT		StringRange
srtout.3.mtu	INTEGER		IntegerRange (0 .. 0)
srtout.3.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtout.3.key	TEXT		StringRange
srtout.3.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.3.stream	TEXT		StringRange
srtout.3.timeout	INTEGER		IntegerRange (0 .. 0)
srtout.3.latency	INTEGER		IntegerRange (0 .. 0)
srtout.3.inputBw	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
srtout.3.maxBw	INTEGER		IntegerRange (0 .. 0)
ipout.4.interface	CHOICE		EnumRange (GIGE-1 GIGE-2)
ipout.4.address	TEXT		StringRange
ipout.4.port	INTEGER		IntegerRange (0 .. 65535)
ipout.4.protocol	CHOICE		EnumRange (RTP UDP)
ipout.4.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.4.address	TEXT		StringRange
srtout.4.port	INTEGER		IntegerRange (0 .. 65535)
srtout.4.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.4.ttl	INTEGER		IntegerRange (1 .. 255)
srtout.4.tos	TEXT		StringRange
srtout.4.mtu	INTEGER		IntegerRange (0 .. 0)
srtout.4.encryption	CHOICE		EnumRange (NONE AES-128 AES-256)
srtout.4.key	TEXT		StringRange
srtout.4.overheadBw	INTEGER		IntegerRange (0 .. 0)
srtout.4.stream	TEXT		StringRange
srtout.4.timeout	INTEGER		IntegerRange (0 .. 0)
srtout.4.latency	INTEGER		IntegerRange (0 .. 0)
srtout.4.inputBw	INTEGER		IntegerRange (0 .. 0)
srtout.4.maxBw	INTEGER		IntegerRange (0 .. 0)
audio.delay	INTEGER		IntegerRange (0 .. 0)
audio.1.enable	CHOICE		EnumRange (OFF ON)
audio.1.samplingRate	CHOICE		EnumRange (PASSTHROUGH 44100Hz 48000Hz 88200Hz 96000Hz)

name	*type*	*flags*	*range*
audio.1.datarate	CHOICE		EnumRange (PASSTHROUGH 8kbps 16kbps 24kbps 32kbps 40kbps 48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 144kbps 160kbps 176kbps 192kbps 224kbps 256kbps 384kbps)
audio.1.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG2-AAC ADTS MPEG4-AAC ADTS MPEG2-AAC LOAS MPEG4-AAC LOAS MPEG1/2 DTS)
audio.1.profile	CHOICE		EnumRange (AAC-LC AAC-HE AAC-HEv2 MPEG1-L1 MPEG1-L2 MPEG1-L3 MPEG2-L1 MPEG2-L2 MPEG2-L3)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.enable	CHOICE		EnumRange (OFF ON)
audio.2.samplingRate	CHOICE		EnumRange (PASSTHROUGH 44100Hz 48000Hz 88200Hz 96000Hz)
audio.2.datarate	CHOICE		EnumRange (PASSTHROUGH 8kbps 16kbps 24kbps 32kbps 40kbps 48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 144kbps 160kbps 176kbps 192kbps 224kbps 256kbps 384kbps)
audio.2.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG2-AAC ADTS MPEG4-AAC ADTS MPEG2-AAC LOAS MPEG4-AAC LOAS MPEG1/2 DTS)
audio.2.profile	CHOICE		EnumRange (AAC-LC AAC-HE AAC-HEv2 MPEG1-L1 MPEG1-L2 MPEG1-L3 MPEG2-L1 MPEG2-L2 MPEG2-L3)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.enable	CHOICE		EnumRange (OFF ON)
audio.3.samplingRate	CHOICE		EnumRange (PASSTHROUGH 44100Hz 48000Hz 88200Hz 96000Hz)

name	*type*	*flags*	*range*
audio.3.datarate	CHOICE		EnumRange (PASSTHROUGH 8kbps 16kbps 24kbps 32kbps 40kbps 48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 144kbps 160kbps 176kbps 192kbps 224kbps 256kbps 384kbps)
audio.3.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG2-AAC ADTS MPEG4-AAC ADTS MPEG2-AAC LOAS MPEG4-AAC LOAS MPEG1/2 DTS)
audio.3.profile	CHOICE		EnumRange (AAC-LC AAC-HE AAC-HEv2 MPEG1-L1 MPEG1-L2 MPEG1-L3 MPEG2-L1 MPEG2-L2 MPEG2-L3)
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.enable	CHOICE		EnumRange (OFF ON)
audio.4.samplingRate	CHOICE		EnumRange (PASSTHROUGH 44100Hz 48000Hz 88200Hz 96000Hz)
audio.4.datarate	CHOICE		EnumRange (PASSTHROUGH 8kbps 16kbps 24kbps 32kbps 40kbps 48kbps 56kbps 64kbps 80kbps 96kbps 112kbps 128kbps 144kbps 160kbps 176kbps 192kbps 224kbps 256kbps 384kbps)
audio.4.coding	CHOICE		EnumRange (DOLBY DIGITAL MPEG2-AAC ADTS MPEG4-AAC ADTS MPEG2-AAC LOAS MPEG4-AAC LOAS MPEG1/2 DTS)
audio.4.profile	CHOICE		EnumRange (AAC-LC AAC-HE AAC-HEv2 MPEG1-L1 MPEG1-L2 MPEG1-L3 MPEG2-L1 MPEG2-L2 MPEG2-L3)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio 2)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 4)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Audio 5)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Audio 6)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Audio 7)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Audio 8)

9.658 TLE-Updater

This page describes the device driver and the device window for the *TLE-Updater* logical device. This device complements the Pass-Scheduler device and reads on request the newest of a set of TLE files, parses this and commands the two TLE lines to an antenna controller device. The device also moves outdated TLE files to an archive directory.

When receiving a "take" command, the device starts to do the following steps. If there occurs a fault in this procedure, the device sets a fault flag indicating this fault and aborts the procedure after the faulty step.

1. Check if a satelliteId has been set. Without a satelliteId a fault is set and the procedure is aborted.
2. Check if the configured source and archive directories exist. If not, a fault is set and the procedure is aborted.
3. Make a list of all files in the source directory matching the pattern *<satelliteId>_YYYYMMDDTHHMMSS.tle* . If no such file exists, a fault is set and the procedure is aborted.
4. If there are multiple files matching the pattern in the source directory, only that one with the newest time stamp in its name is retained, all older/outdated files of this particular satellite are moved to the archive directory.
5. The TLE file is read and the state.* parameters are set from its content. Some simple plausibility checks are done on the file content:
   - The first line (satellite name) must not be longer than 24 characters.
   - The second line must exactly 69 characters long and must start with "1 ".
   - The third line must exactly 69 characters long and must start with "2 ".If one of these checks fails or if the file cannot be read for some reason a fault is set and the procedure is aborted.
6. If everything was OK up to this step, the TLE parameters are set at the antenna controller and after a 100ms delay the antenna controller's "take" command gets issued.

If the procedure failed, the the device shows a fault and the state.* parameters are set to intermediate values which correspond to the procedure step up to which the procedure succeeded.

A "reset" command ("CLR FAULTS" button) may be issued to reset faults and state.* parameters. This is for convenience only, a "reset" is not required as the devices does this anyhow when it starts the procedure the next time with the next "take" command.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all relevant parameters of the device, permits to set the satelliteId and to trigger a "take" manually for testing purposes.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
nameId	The ID of the parameter where to set the satellite name which has been read from the TLE file at the antenna controller.
line1Id	The ID wehre to set the first TLE line at the antenna controller.
line2Id	The ID where to set the second TLE line at the antenna controller.
takeId	The ID of the antenna controller parameter which must be set in order to make the controller "take" the values above.
takeVal	The value which has to be sent with the *takeId* .
tleDir	The directory which gets scanned for TLE files. The directory may be defined as a path relative to the M&C's working directory or as an absolute path starting with a "/" character.
archiveDir	The directory where outdated TLE files are moved to. The directory may be defined as a path relative to the M&C's working directory or as an absolute path starting with a "/" character. Please note, this directory must be at the same physical drive as the *tleDir* because a "rename" command is used to move the files, this does not work over separate drives.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.nameId	TEXT	SAVE SETUP	StringRange
config.line1Id	TEXT	SAVE SETUP	StringRange
config.line2Id	TEXT	SAVE SETUP	StringRange
config.takeId	TEXT	SAVE SETUP	StringRange
config.takeVal	TEXT	SAVE SETUP	StringRange
config.tleDir	TEXT	SAVE SETUP	StringRange
config.archiveDir	TEXT	SAVE SETUP	StringRange
take	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
satelliteId	TEXT	SAVE	StringRange
state.name	TEXT	R/O	StringRange R/O
state.line1	TEXT	R/O	StringRange R/O
state.line2	TEXT	R/O	StringRange R/O
state.filename	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (No satellite ID)
faults.02	ALARM	R/O	AlarmFlagRange R/O (No TLE file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Invalid input dir)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Invalid output dir)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Invalid TLE format)

9.659 TX-Channel-ByChannel

The TX-Channel-ByChannel logical device controls some parameters of an encoder / modulator / power amplifier transmit chain from the satellite database. The device complements the RX-Channel-ByChannel logical device which provides a similar function for receive chains.

TX-Channel-ByChannel supersedes the ZX-Channel device in FILTER-BY-CHANNEL workflow mode, which no longer should be used for new applications. It reads its channel definitions from the SQL based satellite database.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The fault page. There is one fault signal the device (optionally) monitors, this fault is reported as "TX Chain" fault.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

The TX-Channel-ByChannel device follows a 'filter by channel' workflow. You first select a satellite channel from a list, which configures the transmit equipment. The auto-filter mode of the drop down box helps to find the desired satellite channel. Selecting a channel means to set all transmit parameters, the polarization switch and (if configured) also the antenna pointing from the satellite name stored with the selected data record. Optionally, a switch matrix port may be controlled in this mode and the LO frequency may be automatically changed depending on the signal routing.

The TX-Channel-ByChannel device manages a number of transmit parameters, sets these parameters at the encoder, modulator etc. when a channel is selected. The following parameter may be managed by the TX-Channel-ByChannel device:

- txFrequency
- polarization
- symbolrate / bitrate
- dvb mode
- roll off
- pilots
- modulation
- packet size
- mpeg frame size
- video data rate
- video profile (see below)
- video coding format / profile
- nominal eirp
- reduced eirp for lineup
- audio data rate (1)
- audio data rate (2)
- audio data rate (3)

- audio data rate (4)

For an installation of the software not necessarily all of these parameters are available. The TX-Channel-ByChannel device may be configured to ignore some of these parameters.

Please note, that the TX-Channel-ByChannel device neither sets nor clears the TX-ON state of the equipment. The operator is responsible to switch off the transmission before applying a satellite channel data record and also to perform the line up procedure as specified. The LineUp-Control logical device may be used to support the operator with this task. With the LineUp-Control device, the nominal and reduced eirp values from the database are set at this device rather than at the equipment itself. The operator then easily can set the transmit power values required during the line up procedure.

Instead of selecting the satellite and channel you also may enter the transmit parameters at the fields of the device screen. The TX-Channel-ByChannel device forwards these settings to the ancillary devices.

Output switching and LO calculations

The device may control an input port of a combining matrix which routes the signal to a certain transmit chain. With the frequency range and LO frequency data defined in a configuration file, the device only offers the output routes which are valid for the selected transmit frequency. The LO frequency defined for the selected route is applied automatically to the transmit frequency and the frequency setting at the modulator is updated every time the output route changes.

To use this feature, the setup parameter 'config.switchId' must be set to the control parameter of the matrix input to use and a configuration with the frequency range and LO frequencies for all routes must be supplied (the format is described some paragraphs below).

Video profiles

The line settings database contains a parameter called "profile" which controls the basic video encoding mode for the encoder device. This profile contains the MPEG-2 / MPEG-4 selection, the MPEG profile/level and the 4.2.0/4.2.2 color subsampling setting.

Most video encoders don't provide a single parameter to set these values, but use several settings which may differ in name and values they offer to select. To overcome these differences, the TX-Channel-ByChannel device can use special device presets to set the profile parameter in a way individually specialized for the type of video encoder.

To use this feature, store device presets for the profiles you need. Each preset must be named exactly as the profile selection in the channel database (e.g. 'MPEG4-MP@HL-420'). The presets stored for this purpose should be edited in a way, that all parameters which could interfere with other database settings are removed, leaving only the parameters in the preset which set the profile properties.

Then configure the TX-Channel-ByChannel device to route the profile parameter ('profileId' configuration parameter) to the 'applyPreset' parameter of the encoder (e.g. 'ENC.applyPreset'). The TX-Channel-ByChannel device will apply the appropriate device preset

when a channel from the database is selected.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the TX-Channel-ByChannel device with the transmit equipment.

parameter	***description***
switchId	The message ID to set the signal routing at a switch matrix. Defining this adds the signal routing / variable LO frequency capability to the device. This also requires that a configuration file with the frequency range definitions must be supplied and the 'configFile' parameter below must be set to the name of this file.
configFile	Defines the name of the configuration file which defines the frequency range limits and the LO frequencies for each signal path. The ile must be in the satnms home directory and follow the format described in the paragraph below.
frequencyId	The message ID to set the transmit frequency.
fecId	The message ID to set the FEC. If the modulator provides a combined modulation/FEC parameter, leave this field empty.
symbolRateId	The message ID to set the symbol rate. Modulators expecting the symbol rate in 'Msps' and in 'sps' are supported. The symbol rate is only set, if the line settings parameter 'useSymbolRate' is set to SYMBOLRATE.
bitRateId	The message ID to set the bit rate. Modulators expecting the bit rate in 'Mbps' and in 'bps' are supported. The bit rate is only set, if the line settings parameter 'useSymbolRate' is set to BITRATE.
dvbModeId	The message ID to set the DVB-S/DVB-S2 mode.
nlcModeId	The message ID to set the NLC mode.
pilotsId	The message ID to switch pilots ON/OFF.
rollOffId	The message ID to set the roll-off parameter.
modulationId	The message ID to set the modulation scheme (e.g. QPSK)
modulationFecId	Some modulator devices provide a combined modulation FEC parameter instead of separate parameters. For such a device enter the message ID of the combined parameter here and leave the modulationId / fecId fields empty.
packetSizeId	The message ID of the modulator packet size parameter (188/204)

parameter	***description***
frameSizeId	The message ID of the MPEG frame size parameter (short/normal)
videoRateId	The message ID to receive the video data rat (Mbit/s). The device sends this parameter only if th line settings data record has thee video data rat marked as FIXED.
profileId	The message ID of the video coding profile or th 'applyPreset'e parameter of the encoder if the profile shall be set using device presets.
eirpId	The message ID of the parameter to receive the nominal EIRP value for the selected line setting. In most cases this will be the 'nominalEirp' parameter of the LineUp-Control device complementing this device.
reducedEirpId	The message ID of the parameter to receive the reduced (line up) EIRP value for the selected line setting. In most cases this will be the 'reducedEirp' parameter of the LineUp-Control device complementing this device.
audioRateId	The message ID of the encoder audio bitrate parameter (first audio channel).
audio2RateId	The message ID of the encoder audio bitrate parameter (second audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 2 or higher.
audio3RateId	The message ID of the encoder audio bitrate parameter (third audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 4 or higher.
audio4RateId	The message ID of the encoder audio bitrate parameter (fourth audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is to 4.
txFaultId	You may specify the message ID of a fault signal here which is reported as "TX Chain" fault by the device. If you don't want any fault monitoring, leave the field empty.
decoderInputId	The message ID to switch the encoder's output between SAT/IP/ASI
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the encoder.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the encoder.

parameter	***description***
srcIpAddressId	The message ID for setting the IP source or multicast address at the encoder.
srcIpPortId	The message ID for setting the IP source port at the encoder.
dstIpAddressId	The message ID for setting the IP destination address at the IRD.
dstIpPortId	The message ID for setting the IP destination port at the encoder.
ipFecId	The message ID for enabling / disabling the IP FEC
encPhysInterfaceId	The message ID for selecting the physical network interface number at the encoder
txLoFrequency	You may set a BUC LO frequency here, e.g. if the transmit equipment is not capable to compute the modulator output frequency by itself. If your equipment is configured to set the transmit frequency in terms of true RF frequency at the antenna, then leave the field empty or set the LO frequency explicitly to 0. When operated with the 'switchId' set and a frequency range configuration file supplied, LO frequencies are read from this file and the setting of 'txLoFrequency' is ignored.
polSwitchId	The message ID for switching the transmit polarization plane. This may be a waveguide switch position or the 'rxpolplane' parameter of an Antenna-Pointing device.
polSwitchHval	The value to send to the polSwitchId in order to switch the transmit polarization to horizontal. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'V' (the cross polar value).
polSwitchVval	The value to send to the polSwitchId in order to switch the transmit polarization to vertical. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'H' (the cross polar value).
antennaName	The name of the antenna (the Antenna-Pointing device) this transmit chain uses.

Configuration file format

The TX-Channel's configuration file is a plain text file which defines the matrix / signal routing assignments. As this is a system wide assignment, you usually will edit one common configuration file for all TX-Channel-ByChannel devices connected to one output matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes

one matrix input as follows:

column	*contents*	*remark*
1	switch position / matrix output	If the output numbering starts at one or at zero, depends on the matrix / switch device driver.
2	name	The name must not contain spaces, commas o pipe characters. The choices for the 'output' setting are made up of these names.
3	LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
4	low band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	high band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#
#  tx switch matrix example file
#
#  matrix port                  low band edge
#  |    name         LO frequency |         high band edge
#  |    |           |          |         |
   01    UPC1-Ku      13000.0     14000.0    14500.0
   02    UPC2-Ku      13000.0     14000.0    14500.0
   03    UPC3-Ka1      26550.0     27500.0    28250.0
   04    UPC4-Ka2      27150.0     28100.0    28850.0
```

The TX-Channel-ByChannel device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.frequency.min	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.frequency.max	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.bitRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.modulationFecId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.txFaultId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange
config.frameSizeId	TEXT	SAVE SETUP	StringRange
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.profileId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.reducedEirpId	TEXT	SAVE SETUP	StringRange
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audio2RateId	TEXT	SAVE SETUP	StringRange
config.audio3RateId	TEXT	SAVE SETUP	StringRange
config.audio4RateId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.dstIpAddressId	TEXT	SAVE SETUP	StringRange
config.dstIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.ipBufferSizeId	TEXT	SAVE SETUP	StringRange
config.encPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.txLoFrequency	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.polSwitchId	TEXT	SAVE SETUP	StringRange
config.polSwitchHval	TEXT	SAVE SETUP	StringRange
config.polSwitchVval	TEXT	SAVE SETUP	StringRange
config.antennaName	TEXT	SAVE SETUP	StringRange
output	CHOICE		EnumRange DIS (unknown)
antenna	TEXT	R/O	StringRange R/O
satellite	TEXT	R/O	StringRange R/O
fec	CHOICE		EnumRange DIS (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
txFrequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
polarization	CHOICE		EnumRange DIS (X Y)
comment	TEXT	SAVE	StringRange
symbolRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
bitRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange DIS (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange DIS (OFF ON)
rollOff	CHOICE		EnumRange DIS (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
pilots	CHOICE		EnumRange DIS (OFF ON)
modulation	CHOICE		EnumRange DIS (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
packetSize	CHOICE		EnumRange DIS (188 204)
frameSize	CHOICE		EnumRange DIS (SHORT NORMAL)
videoRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
profile	CHOICE		EnumRange DIS (UNKNOWN)
eirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
reducedEirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
audioRate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio2Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio3Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio4Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
decoderInput	CHOICE		EnumRange (SAT IP ASI)

name	*type*	*flags*	*range*
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
dstIpAddress	TEXT		StringRange
dstIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
ipBufferSize	INTEGER		IntegerRange (0 .. 0)
encPhysInterface	INTEGER		IntegerRange (1 .. 4)
reloadProfiles	TEXT		StringRange
channel	CHOICE		EnumRange (unknown)
reset	TEXT		StringRange
locked	BOOLEAN		BooleanRange (true false)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX Chain)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.660 TX-Channel-BySatellite

The TX-Channel-BySatellite logical device controls some parameters of an encoder / modulator / power amplifier transmit chain from the satellite database. The device complements the RX-Channel-BySatellite logical device which provides a similar function for receive chains.

TX-Channel-BySatellite supersedes the TX-Channel device in FILTER-BY-SATELLITE workflow mode. It reads its channel definitions from the SQL based satellite database.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The fault page. There is one fault signal the device (optionally) monitors, this fault is reported as "TX Chain" fault.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation

The device follows a 'filter by satellite' workflow, here the 'channel' field only shows the channels of the satellite the antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the TX-Channel's device screen may me used or the antenna may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. This operation mode is only possible in M&C environments.

The device provides a locking mechanism which permits to lock the settings of the transmit chain and the pointing of the antenna while the transmission is running. The locking mechanism is coupled to that one of RX-Channel-BySatellite devices. This means, locking the antenna for a transmission also locks the antenna pointing for all RX-Channel-BySatellite devices.

The TX-Channel-BySatellite device manages a number of transmit parameters, sets these parameters at the encoder, modulator etc. when a channel is selected. The following parameter may be managed by the TX-Channel-BySatellite device:

- txFrequency
- polarization
- symbolrate / bitrate
- dvb mode
- roll off
- pilots
- modulation
- packet size
- mpeg frame size
- video data rate
- video profile (see below)
- video coding format / profile
- nominal eirp
- reduced eirp for lineup
- audio data rate (1)
- audio data rate (2)

- audio data rate (3)
- audio data rate (4)

For an installation of the software not necessarily all of these parameters are available. The TX-Channel-BySatellite device may be configured to ignore some of these parameters.

Please note, that the TX-Channel-BySatellite device neither sets nor clears the TX-ON state of the equipment. The operator is responsible to switch off the transmission before applying a satellite channel data record and also to perform the line up procedure as specified. The LineUp-Control logical device may be used to support the operator with this task. With the LineUp-Control device, the nominal and reduced eirp values from the database are set at this device rather than at the equipment itself. The operator then easily can set the transmit power values required during the line up procedure.

Instead of selecting the satellite and channel you also may enter the transmit parameters at the fields of the device screen. The TX-Channel-BySatellite device forwards these settings to the ancillary devices.

Video profiles

The line settings database contains a parameter called "profile" which controls the basic video encoding mode for the encoder device. This profile contains the MPEG-2 / MPEG-4 selection, the MPEG profile/level and the 4.2.0/4.2.2 color subsampling setting.

Most video encoders don't provide a single parameter to set these values, but use several settings which may differ in name and values they offer to select. To overcome these differences, the TX-Channel-BySatellite device can use special device presets to set the profile parameter in a way individually specialized for the type of video encoder.

To use this feature, store device presets for the profiles you need. Each preset must be named exactly as the profile selection in the channel database (e.g. 'MPEG4-MP@HL-420'). The presets stored for this purpose should be edited in a way, that all parameters which could interfere with other database settings are removed, leaving only the parameters in the preset which set the profile properties.

Then configure the TX-Channel-BySatellite device to route the profile parameter ('profileId' configuration parameter) to the 'applyPreset' parameter of the encoder (e.g. 'ENC.applyPreset'). The TX-Channel-BySatellite device will apply the appropriate device preset when a channel from the database is selected.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the TX-Channel-BySatellite device with the transmit equipment.

parameter	***description***

parameter	***description***
switchId	The message ID to set the signal routing at a switch matrix. Defining this adds the signal routing / variable LO frequency capability to the device in 'workflowMode=FILTER-BY-CHANNEL' mode. This also requires that a configuration file with the frequency range definitions must be supplied and the 'configFile' parameter below must be set to the name of this file. With 'workflowMode=FILTER-BY-SATELLITE' the 'switchId' parameter must remain empty.
configFile	Defines the name of the configuration file which defines the frequency range limits and the LO frequencies for each signal path. The ile must be in the satnms home directory and follow the format described in the paragraph below.
frequencyId	The message ID to set the transmit frequency.
fecId	The message ID to set the FEC. If the modulator provides a combined modulation/FEC parameter, leave this field empty.
symbolRateId	The message ID to set the symbol rate. Modulators expecting the symbol rate in 'Msps' and in 'sps' are supported. The symbol rate is only set, if the line settings parameter 'useSymbolRate' is set to SYMBOLRATE.
bitRateId	The message ID to set the bit rate. Modulators expecting the bit rate in 'Mbps' and in 'bps' are supported. The bit rate is only set, if the line settings parameter 'useSymbolRate' is set to BITRATE.
dvbModeId	The message ID to set the DVB-S/DVB-S2 mode.
nlcModeId	The message ID to set the NLC mode.
pilotsId	The message ID to switch pilots ON/OFF.
rollOffId	The message ID to set the roll-off parameter.
modulationId	The message ID to set the modulation scheme (e.g. QPSK)
modulationFecId	Some modulator devices provide a combined modulation FEC parameter instead of separate parameters. For such a device enter the message ID of the combined parameter here and leave the modulationId / fecId fields empty.
packetSizeId	The message ID of the modulator packet size parameter (188/204)
frameSizeId	The message ID of the MPEG frame size parameter (short/normal)
videoRateId	The message ID to receive the video data rat (Mbit/s). The device sends this parameter only if th line settings data record has thee video data rat marked as FIXED.

parameter	***description***
profileId	The message ID of the video coding profile or th 'applyPreset'e parameter of the encoder if the profile shall be set using device presets.
eirpId	The message ID of the parameter to receive the nominal EIRP value for the selected line setting. In most cases this will be the 'nominalEirp' parameter of the LineUp-Control device complementing this device.
reducedEirpId	The message ID of the parameter to receive the reduced (line up) EIRP value for the selected line setting. In most cases this will be the 'reducedEirp' parameter of the LineUp-Control device complementing this device.
audioRateId	The message ID of the encoder audio bitrate parameter (first audio channel).
audio2RateId	The message ID of the encoder audio bitrate parameter (second audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 2 or higher.
audio3RateId	The message ID of the encoder audio bitrate parameter (third audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 4 or higher.
audio4RateId	The message ID of the encoder audio bitrate parameter (fourth audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is to 4.
txFaultId	You may specify the message ID of a fault signal here which is reported as "TX Chain" fault by the device. If you don't want any fault monitoring, leave the field empty.
decoderInputId	The message ID to switch the encoder's output between SAT/IP/ASI
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the encoder.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the encoder.
srcIpAddressId	The message ID for setting the IP source or multicast address at the encoder.
srcIpPortId	The message ID for setting the IP source port at the encoder.
dstIpAddressId	The message ID for setting the IP destination address at the IRD.
dstIpPortId	The message ID for setting the IP destination port at the encoder.

parameter	***description***
ipFecId	The message ID for enabling / disabling the IP FEC
encPhysInterfaceId	The message ID for selecting the physical network interface number at the encoder
txLoFrequency	You may set a BUC LO frequency here, e.g. if the transmit equipment is not capable to compute the modulator output frequency by itself. If your equipment is configured to set the transmit frequency in terms of true RF frequency at the antenna, then leave the field empty or set the LO frequency explicitly to 0. When operated with the 'switchId' set and a frequency range configuration file supplied, LO frequencies are read from this file and the setting of 'txLoFrequency' is ignored.
polSwitchId	The message ID for switching the transmit polarization plane. This may be a waveguide switch position or the 'rxpolplane' parameter of an Antenna-Pointing device.
polSwitchHval	The value to send to the polSwitchId in order to switch the transmit polarization to horizontal. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'V' (the cross polar value).
polSwitchVval	The value to send to the polSwitchId in order to switch the transmit polarization to vertical. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'H' (the cross polar value).
antennaName	The name of the antenna (the Antenna-Pointing device) this transmit chain uses.
lockByIP	Setting this to YES forces the device to require a lock to be released from the same computer from where it was set. With the other operation modes 'lockByIP' is ignored.

Configuration file format

The TX-Channel's configuration file is a plain text file which defines the matrix / signal routing assignments. As this is a system wide assignment, you usually will edit one common configuration file for all TX-Channel-BySatellite devices connected to one output matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	***contents***	***remark***

column	*contents*	*remark*
1	switch position / matrix output	If the output numbering starts at one or at zero, depends on the matrix / switch device driver.
2	name	The name must not contain spaces, commas o pipe characters ('|'). The choices for the 'output' setting are made up of these names.
3	LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
4	low band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	high band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#
#  tx switch matrix example file
#
#  matrix port                  low band edge
#  |    name         LO frequency |         high band edge
#  |    |          |         |         |
   01    UPC1-Ku      13000.0     14000.0    14500.0
   02    UPC2-Ku      13000.0     14000.0    14500.0
   03    UPC3-Ka1     26550.0     27500.0    28250.0
   04    UPC4-Ka2     27150.0     28100.0    28850.0
```

The TX-Channel-BySatellite device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.frequency.min	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.frequency.max	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.bitRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.modulationFecId	TEXT	SAVE SETUP	StringRange
config.txFaultId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange
config.frameSizeId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.profileId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.reducedEirpId	TEXT	SAVE SETUP	StringRange
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audio2RateId	TEXT	SAVE SETUP	StringRange
config.audio3RateId	TEXT	SAVE SETUP	StringRange
config.audio4RateId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.dstIpAddressId	TEXT	SAVE SETUP	StringRange
config.dstIpPortId	TEXT	SAVE SETUP	StringRange
config.ipFecId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ipBufferSizeId	TEXT	SAVE SETUP	StringRange
config.encPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.txLoFrequency	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.polSwitchId	TEXT	SAVE SETUP	StringRange
config.polSwitchHval	TEXT	SAVE SETUP	StringRange
config.polSwitchVval	TEXT	SAVE SETUP	StringRange
config.antennaName	TEXT	SAVE SETUP	StringRange
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
antenna	TEXT	R/O	StringRange R/O
satellite	CHOICE		EnumRange (unknown)
fec	CHOICE		EnumRange DIS (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
txFrequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
polarization	CHOICE		EnumRange DIS (X Y)
comment	TEXT	SAVE	StringRange
symbolRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
bitRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange DIS (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
rollOff	CHOICE		EnumRange DIS (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
pilots	CHOICE		EnumRange DIS (OFF ON)
modulation	CHOICE		EnumRange DIS (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
packetSize	CHOICE		EnumRange DIS (188 204)
frameSize	CHOICE		EnumRange DIS (SHORT NORMAL)
videoRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
profile	CHOICE		EnumRange DIS (UNKNOWN)
eirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
reducedEirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
audioRate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio2Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio3Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio4Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
decoderInput	CHOICE		EnumRange (SAT IP ASI)
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
dstIpAddress	TEXT		StringRange
dstIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
ipBufferSize	INTEGER		IntegerRange (0 .. 0)
encPhysInterface	INTEGER		IntegerRange (1 .. 4)
reloadProfiles	TEXT		StringRange
channel	CHOICE		EnumRange (unknown)
reset	TEXT		StringRange
locked	BOOLEAN	R/O	BooleanRange R/O (true false)
satLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
polLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
unsupportedPol	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedFreq	BOOLEAN	R/O	BooleanRange R/O (true false)
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX Chain)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Database access)

9.661 TX-Channel

The TX-Channel logical device controls some parameters of an encoder / modulator / power amplifier transmit chain from a common channel database. The device complements the RX-Channel logical device which provides a similar function for receive chains.

TX-Channel supersedes the TX-LineSettings-CDB device, which no longer should be used for new applications. The TX-Channel device supports a larger number of parameters, may be used in M&C and in NMS environments and it supports an alternative operation mode / work flow.

The name TX-Channel reflects that the set of parameters managed by the device extend the standard channel parameters.

Heart of the TX-Channel device is a central database (CDB) storing the line settings for all satellite channels used in the system. In a NMS environment the database is located in the NMS server, providing an synchronization mechanism which creates local (read-only) copies of the database for offline operation of the VLCs. In a NMS environment, the sat-nms client software looks up the channel lists directly at the NMS database server which significantly reduces the amount of data which has to be exchanged between NMS and VLC. In a M&C environment the CDB is located at the M&C server, no synchronization is required in this case.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The channel database editor page.
- --- The fault page. There is one fault signal the device (optionally) monitors, this fault is reported as "TX Chain" fault.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

The TX-Channel device supports two different operation modes / work flow settings which are selected by the 'workflowMode' configuration parameter.

With 'workflowMode=FILTER-BY-CHANNEL' (the default mode) the 'channel' field always shows all line settings data records. The auto-filter mode of the drop down box helps to find the desired satellite channel. Selecting a channel means to set all transmit parameters, the polarization switch and (if configured) also the antenna pointing from the satellite name stored with the selected data record. This operation mode may be used in M&C and NMS environments as well. Optionally, a switch matrix port may be controlled in this mode and the LO frequency may be automatically changed depending on the signal routing.

With 'workflowMode=FILTER-BY-SATELLITE', the 'channel' field only shows the channels of the satellite the antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the TX-Channel's device screen may me used or the antenna may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. This operation mode is only possible in M&C environments.

The FILTER-BY-SATELLITE mode provides a locking mechanism which permits to lock the settings of the transmit chain and the pointing of the antenna while the transmission is running. The locking mechanism is coupled to that one of RX-Channel devices working in FILTER-BY-SATELLITE mode as well. This means, locking the antenna for a transmission also

locks the antenna pointing for all RX-Channel devices.

The TX-Channel device manages a number of transmit parameters, sets these parameters at the encoder, modulator etc. when a channel is selected. The following parameter may be managed by the TX-Channel device:

- txFrequency
- polarization
- symbolrate / bitrate
- dvb mode
- roll off
- pilots
- modulation
- packet size
- mpeg frame size
- video data rate
- video profile (see below)
- video coding format / profile
- nominal eirp
- reduced eirp for lineup
- audio data rate (1)
- audio data rate (2)
- audio data rate (3)
- audio data rate (4) For an installation of the software not necessarily all of these parameters are available. The TX-Channel device may be configured to ignore some of these parameters.

Please note, that the TX-Channel device neither sets nor clears the TX-ON state of the equipment. The operator is responsible to switch off the transmission before applying a satellite channel data record and also to perform the line up procedure as specified. The LineUp-Control logical device may be used to support the operator with this task. With the LineUp-Control device, the nominal and reduced eirp values from the database are set at this device rather than at the equipment itself. The operator then easily can set the transmit power values required during the line up procedure.

Instead of selecting the satellite and channel you also may enter the transmit parameters at the fields of the device screen. The TX-Channel device forwards these settings to the ancillary devices.

Output switching and LO calculations

When operated in 'workflowMode=FILTER-BY-CHANNEL', the device may control an input port of a combining matrix which routes the signal to a certain transmit chain. With the frequency range and LO frequency data defined in a configuration file, the device only offers the output routes which are valid for the selected transmit frequency. The LO frequency defined for the selected route is applied automatically to the transmit frequency and the frequency setting at the modulator is updated every time the output route changes.

To use this feature, the setup parameter 'config.switchId' must be set to the control parameter

of the matrix input to use and a configuration with the frequency range and LO frequencies for all routes must be supplied (the format is described some paragraphs below).

Video profiles

The line settings database contains a parameter called "profile" which controls the basic video encoding mode for the encoder device. This profile contains the MPEG-2 / MPEG-4 selection, the MPEG profile/level and the 4.2.0/4.2.2 color subsampling setting.

Most video encoders don't provide a single parameter to set these values, but use several settings which may differ in name and values they offer to select. To overcome these differences, the TX-Channel device can use special device presets to set the profile parameter in a way individually specialized for the type of video encoder.

To use this feature, store device presets for the profiles you need. Each preset must be named exactly as the profile selection in the channel database (e.g. 'MPEG4-MP@HL-420'). The presets stored for this purpose should be edited in a way, that all parameters which could interfere with other database settings are removed, leaving only the parameters in the preset which set the profile properties.

Then configure the TX-Channel device to route the profile parameter ('profileId' configuration parameter) to the 'applyPreset' parameter of the encoder (e.g. 'ENC.applyPreset'). The TX-Channel device will apply the appropriate device preset when a channel from the database is selected.

Operation (database edit)

With the second page of the device window of the TX-Channel device channel definitions are edited, copied or deleted. All TX-Channel devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the TX-Channel device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the equipment settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'
- Delete a channel:

1. Select the channel to delete
2. Click 'DELETE'

Local database copy

The following paragraph only applies to NMS installations. M&C installation manage their CDB locally at the M&C server and do not require a local copy of the data.

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the TX-Channel functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a TX-Channel device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the TX-Channel device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS

server, the TX-Channel device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the TX-Channel device with the transmit equipment.

parameter	***description***
switchId	The message ID to set the signal routing at a switch matrix. Defining this adds the signal routing / variable LO frequency capability to the device in 'workflowMode=FILTER-BY-CHANNEL' mode. This also requires that a configuration file with the frequency range definitions must be supplied and the 'configFile' parameter below must be set to the name of this file. With 'workflowMode=FILTER-BY-SATELLITE' the 'switchId' parameter must remain empty.
configFile	Defines the name of the configuration file which defines the frequency range limits and the LO frequencies for each signal path. The ile must be in the satnms home directory and follow the format described in the paragraph below.
frequencyId	The message ID to set the transmit frequency.
fecId	The message ID to set the FEC. If the modulator provides a combined modulation/FEC parameter, leave this field empty.
symbolRateId	The message ID to set the symbol rate. Modulators expecting the symbol rate in 'Msps' and in 'sps' are supported. The symbol rate is only set, if the line settings parameter 'useSymbolRate' is set to SYMBOLRATE.
bitRateId	The message ID to set the bit rate. Modulators expecting the bit rate in 'Mbps' and in 'bps' are supported. The bit rate is only set, if the line settings parameter 'useSymbolRate' is set to BITRATE.
dvbModeId	The message ID to set the DVB-S/DVB-S2 mode.
pilotsId	The message ID to switch pilots ON/OFF.
rollOffId	The message ID to set the roll-off parameter.
modulationId	The message ID to set the modulation scheme (e.g. QPSK)
modulationFecId	Some modulator devices provide a combined modulation FEC parameter instead of separate parameters. For such a device enter the message ID of the combined parameter here and leave the modulationId / fecId fields empty.

parameter	***description***
packetSizeId	The message ID of the modulator packet size parameter (188/204)
frameSizeId	The message ID of the MPEG frame size parameter (short/normal)
videoRateId	The message ID to receive the video data rat (Mbit/s). The device sends this parameter only if th line settings data record has thee video data rat marked as FIXED.
profileId	The message ID of the video coding profile or th 'applyPreset'e parameter of the encoder if the profile shall be set using device presets.
eirpId	The message ID of the parameter to receive the nominal EIRP value for the selected line setting. In most cases this will be the 'nominalEirp' parameter of the LineUp-Control device complementing this device.
reducedEirpId	The message ID of the parameter to receive the reduced (line up) EIRP value for the selected line setting. In most cases this will be the 'reducedEirp' parameter of the LineUp-Control device complementing this device.
audioRateId	The message ID of the encoder audio bitrate parameter (first audio channel).
audio2RateId	The message ID of the encoder audio bitrate parameter (second audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 2 or higher.
audio3RateId	The message ID of the encoder audio bitrate parameter (third audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 4 or higher.
audio4RateId	The message ID of the encoder audio bitrate parameter (fourth audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is to 4.
txFaultId	You may specify the message ID of a fault signal here which is reported as "TX Chain" fault by the device. If you don't want any fault monitoring, leave the field empty.
decoderInputId	The message ID to switch the encoder's output between SAT/IP/ASI
ipTypeId	The message ID for selecting the IP protocol family (UNICAST/MULTICAST) at the encoder.
ipProtocolId	The message ID for selecting the IP protocol (UDP, RTP, etc.) at the encoder.

parameter	***description***
srcIpAddressId	The message ID for setting the IP source or multicast address at the encoder.
srcIpPortId	The message ID for setting the IP source port at the encoder.
dstIpAddressId	The message ID for setting the IP destination address at the IRD.
dstIpPortId	The message ID for setting the IP destination port at the encoder.
ipFecId	The message ID for enabling / disabling the IP FEC
encPhysInterfaceId	The message ID for selecting the physical network interface number at the encoder
txLoFrequency	You may set a BUC LO frequency here, e.g. if the transmit equipment is not capable to compute the modulator output frequency by itself. If your equipment is configured to set the transmit frequency in terms of true RF frequency at the antenna, then leave the field empty or set the LO frequency explicitly to 0. When operated with the 'switchId' set and a frequency range configuration file supplied, LO frequencies are read from this file and the setting of 'txLoFrequency' is ignored.
polSwitchId	The message ID for switching the transmit polarization plane. This may be a waveguide switch position or the 'rxpolplane' parameter of an Antenna-Pointing device.
polSwitchHval	The value to send to the polSwitchId in order to switch the transmit polarization to horizontal. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'V' (the cross polar value).
polSwitchVval	The value to send to the polSwitchId in order to switch the transmit polarization to vertical. Please note, that when using the 'rxpolplane' parameter of an Antenna-Pointing device, you have to set this to 'H' (the cross polar value).
antennaName	The name of the antenna (the Antenna-Pointing device) this transmit chain uses.
workflowMode	Selects the workflow / operation mode of the device. The operation mode FILTER-BY-SATELLITE is only possible in M&C environments.
lockByIP	Setting this to YES forces the device in FILTER-BY-SATELLITE mode to require a lock to be released from the same computer from where it was set. With the other operation modes 'lockByIP' is ignored.

Configuration file format

The TX-Channel's configuration file is a plain text file which defines the matrix / signal routing assignments. As this is a system wide assignment, you usually will edit one common configuration file for all TX-Channel devices connected to one output matrix.

In the file all empty lines and lines starting with a '#' character are ignored. All other lines are treated as table rows where the cells are separated by space characters. Each row describes one matrix input as follows:

column	***contents***	***remark***
1	switch position / matrix output	If the output numbering starts at one or at zero, depends on the matrix / switch device driver.
2	name	The name must not contain spaces, commas o pipe characters ('
3	LO frequency for this port	The LO frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
4	low band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)
5	high band edge for this port	The frequency must be given as a decimal number (MHz) with a decimal point (no commas, no thousand's separators are allowed)

Example:

```
#
#  tx switch matrix example file
#
#  matrix port                  low band edge
#  |    name         LO frequency |         high band edge
#  |    |          |         |         |
   01   UPC1-Ku      13000.0     14000.0    14500.0
   02   UPC2-Ku      13000.0     14000.0    14500.0
   03   UPC3-Ka1      26550.0     27500.0    28250.0
   04   UPC4-Ka2      27150.0     28100.0    28850.0
```

The TX-Channel device parses this file on startup and when it receives a new name for the configuration file. So when you change the configuration file in the running M&C system, the best way is to restart the M&C server process to make all devices read the configuration file again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.configFile	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.bitRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.modulationFecId	TEXT	SAVE SETUP	StringRange
config.txFaultId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.frameSizeId	TEXT	SAVE SETUP	StringRange
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.profileId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.reducedEirpId	TEXT	SAVE SETUP	StringRange
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audio2RateId	TEXT	SAVE SETUP	StringRange
config.audio3RateId	TEXT	SAVE SETUP	StringRange
config.audio4RateId	TEXT	SAVE SETUP	StringRange
config.decoderInputId	TEXT	SAVE SETUP	StringRange
config.ipTypeId	TEXT	SAVE SETUP	StringRange
config.ipProtocolId	TEXT	SAVE SETUP	StringRange
config.srcIpAddressId	TEXT	SAVE SETUP	StringRange
config.srcIpPortId	TEXT	SAVE SETUP	StringRange
config.dstIpAddressId	TEXT	SAVE SETUP	StringRange
config.dstIpPortId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.ipFecId	TEXT	SAVE SETUP	StringRange
config.ipBufferSizeId	TEXT	SAVE SETUP	StringRange
config.encPhysInterfaceId	TEXT	SAVE SETUP	StringRange
config.txLoFrequency	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.polSwitchId	TEXT	SAVE SETUP	StringRange
config.polSwitchHval	TEXT	SAVE SETUP	StringRange
config.polSwitchVval	TEXT	SAVE SETUP	StringRange
config.antennaName	TEXT	SAVE SETUP	StringRange
config.workflowMode	CHOICE	SAVE SETUP	EnumRange (FILTER-BY-CHANNEL FILTER-BY-SATELLITE)
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
output	CHOICE		EnumRange DIS (unknown)
antenna	TEXT	R/O	StringRange R/O
satellite	TEXT		StringRange DIS
fec	CHOICE		EnumRange DIS (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
txFrequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
polarization	CHOICE		EnumRange DIS (X Y)
comment	TEXT	SAVE	StringRange
symbolRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
bitRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange DIS (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange DIS (OFF ON)
rollOff	CHOICE		EnumRange DIS (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
pilots	CHOICE		EnumRange DIS (OFF ON)
modulation	CHOICE		EnumRange DIS (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
packetSize	CHOICE		EnumRange DIS (188 204)
frameSize	CHOICE		EnumRange DIS (SHORT NORMAL)
videoRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
profile	CHOICE		EnumRange DIS (UNKNOWN)
eirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
reducedEirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
audioRate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio2Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio3Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio4Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
decoderInput	CHOICE		EnumRange (SAT IP ASI)

name	*type*	*flags*	*range*
ipType	CHOICE		EnumRange (UNICAST MULTICAST)
ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
srcIpAddress	TEXT		StringRange
srcIpPort	INTEGER		IntegerRange (0 .. 65535)
dstIpAddress	TEXT		StringRange
dstIpPort	INTEGER		IntegerRange (0 .. 65535)
ipFec	CHOICE		EnumRange (OFF ON)
ipBufferSize	INTEGER		IntegerRange (0 .. 0)
encPhysInterface	INTEGER		IntegerRange (1 .. 4)
reloadProfiles	TEXT		StringRange
channel	CHOICE		EnumRange (@@@channels)
reset	TEXT		StringRange
locked	BOOLEAN		BooleanRange (true false)
satLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
polLocked	BOOLEAN	R/O	BooleanRange R/O (true false)
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
unsupportedPol	BOOLEAN	R/O	BooleanRange R/O (true false)
unsupportedFreq	BOOLEAN	R/O	BooleanRange R/O (true false)
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	CHOICE		EnumRange (NONE)
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.bitRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.useSymbolRate	CHOICE		EnumRange (SYMBOLRATE BITRATE)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
edit.packetSize	CHOICE		EnumRange (188 204)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.nlcMode	CHOICE		EnumRange (OFF ON)
edit.frameSize	CHOICE		EnumRange (SHORT NORMAL)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.videoRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.videoRateAuto	CHOICE		EnumRange (FIXED AUTO)

name	*type*	*flags*	*range*
edit.profile	CHOICE		EnumRange (MPEG2-MP@ML-420 MPEG2-MP@LL-420 MPEG2-HP@ML-422 MPEG2-HP@LL-422 MPEG2-SP@LL-420 MPEG2-SP@ML-420 MPEG2-MP@H14-420 MPEG2-MP@HL-420 MPEG2-HP@H14-422 MPEG2-HP@HL-422 MPEG2-HP@HL-420 MPEG2-HP@H14-420 MPEG2-HP@ML-420 MPEG4-BP@L2-420 MPEG4-BP@L3-420 MPEG4-BP@L4-420 MPEG4-MP@L3-420 MPEG4-MP@L4-420 MPEG4-MP@L4.2-420 MPEG4-HP@L3-420 MPEG4-HP@L4-420 MPEG4-HP@L4.2-420 MPEG4-HP@L3-422 MPEG4-HP@L4-422 MPEG4-HP@L4.2-422)
edit.eirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.reducedEirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.autoLineUp	CHOICE		EnumRange DIS (OFF ON)
edit.audioRate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio2Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio3Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio4Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audioChannels	CHOICE		EnumRange (1 2 3 4)
edit.comment	TEXT		StringRange
edit.user	TEXT		StringRange

name	*type*	*flags*	*range*
edit.programTitle	TEXT		StringRange
edit.decoderInput	CHOICE		EnumRange (SAT IP ASI)
edit.ipType	CHOICE		EnumRange (UNICAST MULTICAST)
edit.ipProtocol	CHOICE		EnumRange (UDP RTP TCP ZIXI)
edit.srcIpAddress	TEXT		StringRange
edit.srcIpPort	INTEGER		IntegerRange (0 .. 65535)
edit.dstIpAddress	TEXT		StringRange
edit.dstIpPort	INTEGER		IntegerRange (0 .. 65535)
edit.ipFec	CHOICE		EnumRange (OFF ON)
edit.ipBufferSize	INTEGER		IntegerRange (0 .. 0)
edit.encPhysInterface	INTEGER		IntegerRange (1 .. 4)
edit.decPhysInterface	INTEGER		IntegerRange (1 .. 4)
edit.satEnable	TEXT	R/O	StringRange R/O
edit.ipEnable	TEXT	R/O	StringRange R/O
edit.brEnable	TEXT	R/O	StringRange R/O
edit.srEnable	TEXT	R/O	StringRange R/O
edit.vrEnable	TEXT	R/O	StringRange R/O
edit.ar2Enable	TEXT	R/O	StringRange R/O
edit.ar3Enable	TEXT	R/O	StringRange R/O
edit.ar4Enable	TEXT	R/O	StringRange R/O
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
eventDescription	TEXT	SAVE	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX Chain)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Config file)

9.662 TX-ChannelSelect-CDB

The TX-ChannelSelect-CDB logical device controls some basic parameters of an encoder / modulator from a common channel database. The device complements the RX-ChannelSelect-CDB logical device which provides a similar function for receive chains.

TX-ChannelSelect-CDB is comparable to the TX-ChannelSelect device, but is intended for the use in a sat-nms NMS system. "CDB" in the driver name stands for "Central Data Base", underlining that all TX-ChannelSelect-CDB instances use one central satellite channel database located at the NMS server. The TX-ChannelSelect-CDB device shares this channel database with it's receive chain pendant.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The channel database editor page.
- --- The fault page. There is one fault signal the device (optionally) monitors, this fault is reported as "TX Chain" fault.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

To configure the transmit chain for a certain satellite channel, simply select a satellite channel from the list. This commands the parameters

- tx frequency
- fec
- symbol rate
- dvb mode
- pilots
- roll-off

- modulation
- tx polarization

as stored with the channel definition. Please note, that the TX-ChannelSelect-CDB device neither controls the TX-ON state if the equipment nor the transmit power. The operator is responsible to switch off the transmission before applying a satellite channel data record and also to perform the line up procedure as specified.

Instead of selecting the satellite and channel you also may enter the transmit parameters at the frequency / pol / symbol rate / fec fields. The TX-Channelselect-CDB device forwards these settings to the ancillary devices.

Operation (database edit)

With the second page of the device window of the TX-ChannelSelect-CDB device channel definitions are edited, copied or deleted. All TX-ChannelSelect devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the TX-ChannelSelect-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the receiver settings.

- Change certain parameters of a channel:
    1. Select the channel to modify
    2. Edit the settings
    3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore characters when the channel gets stored.
- Save the actual receive settings as a new channel:
    1. Click 'READ' to fill the fields from actual receive settings.
    2. Optionally complete the transmit parameters for the channel.
    3. Enter the desired satellite / channel name.
    4. Click 'SAVE'
- Delete a channel:
    1. Select the channel to delete
    2. Click 'DELETE'

Local database copy

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the TX-ChannelSelect-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a TX-ChannelSelect-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the TX-ChannelSelect-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the TX-ChannelSelect-CDB device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the TX-ChannelSelect-CDB device with the transmit equipment.

parameter	***description***
frequencyId	The message ID to set the transmit frequency.

parameter	*description*
fecId	The message ID to set the FEC. If the modulator provides a combined modulation/FEC parameter, leave this field empty.
symbolRateId	The message ID to set the symbol rate. Modulators expecting the symbol rate in 'Msps' and in 'sps' are supported.
dvbModeId	The message ID to set the DVB-S/DVB-S2 mode.
pilotId	The message ID to switch pilots ON/OFF.
rollOffId	The message ID to set the roll-off parameter.
modulationId	The message ID to set the modulation scheme (e.g. QPSK)
modulationFecId	Some modulator devices provide a combined modulation FEC parameter instead of separate parameters. For such a device enter the message ID of the combined parameter here and leave the modulationId / fecId fields empty.
txFaultId	You may specify the message ID of a fault signal here which is reported as "TX Chain" fault by the device. If you don't want any fault monitoring, leave the field empty.
polarizationId	The message ID to set the transmit polarization. The device sends X / Y from the channel definition to this ID. With a waveguide switch for pol select you may us a Choice-Translator logical device to translate X/Y into the proper switch positions.
txLoFrequency	You may set a BUC LO frequency here, e.g. if the transmit equipment is not capable to compute the modulator output frequency by itself. If your equipment is configured to set the transmit frequency in terms of true RF frequency at the antenna, then leave the field empty or set the LO frequency explicitly to 0.
txLoPolarity	Set this to "NORM" if the LO frequency is *below* RF transmit frequency, "INVERT" if the LO is *above* transmit frequency.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

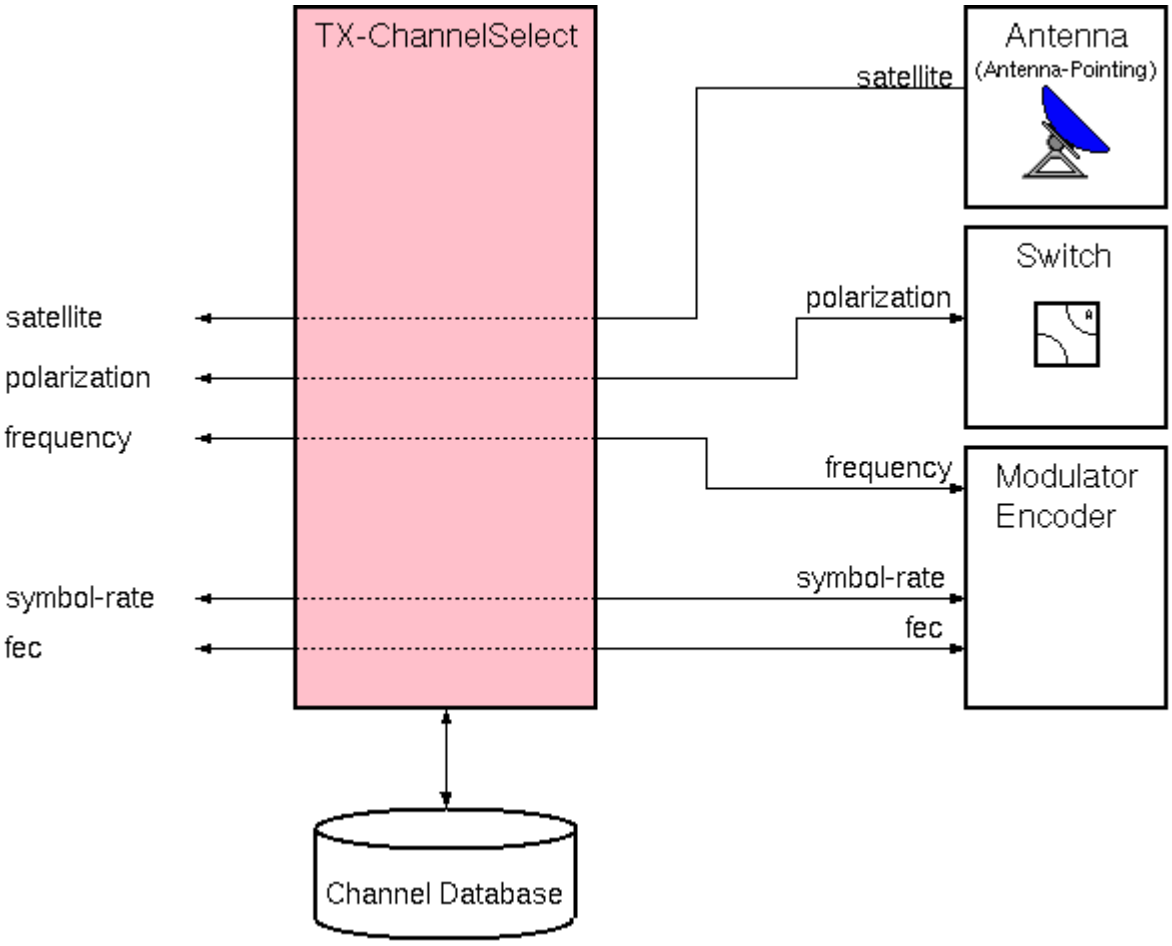
name	*type*	*flags*	*range*
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.nlcModeId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.modulationFecId	TEXT	SAVE SETUP	StringRange
config.txFaultId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.txLoFrequency	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.txLoPolarity	CHOICE	SAVE SETUP	EnumRange (NORM INVERT)
fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
txFrequency	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	CHOICE		EnumRange (X Y)
comment	TEXT	SAVE	StringRange
symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
nlcMode	CHOICE		EnumRange (OFF ON)
rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
pilots	CHOICE		EnumRange (OFF ON)
modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK)
channel	CHOICE		EnumRange (@@@channels)
reset	TEXT		StringRange
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.nlcMode	CHOICE		EnumRange (OFF ON)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.comment	TEXT		StringRange
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange

name	*type*	*flags*	*range*
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX Chain)

9.663 TX-ChannelSelect

The TX-ChannelSelect logical device controls some basic parameters of an encoder / modulator from a common channel database. The device complements the RX-ChannelSelect logical device which provides the same (and more) functions for receive chains.



The TX-ChannelSelect device simply routes the user level parameters 'satellite', 'polarization' and 'frequency' to the physical device which are configured for the particular TX-ChannelSelect

device. The TX-ChannelSelect device is definitely assigned to one antenna, this is one major difference to the RX-ChannelSelect device which supports a signal selection from multiple antennas.

Beside the function described above, the TX-ChannelSelect device implements a locking mechanism which prevents users from changing parameters at a transmit chain which is currently in use. This protection mechanism also works together with RX-ChannelSelect devices, in fact antenna pointing as a common resource are locked for both types of devices at the same time.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (channel select) page for the device.
- --- The channel database editor page.
- --- The fault page. The device provides four configurable fault inputs. Unconfigured faults appear as 'Fault 01' .. 'Fault 04'. A configured fault inherits the description from the fault it refers to.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

Selecting an already defined satellite channels is a simple 2-step procedure:

1. Select the satellite you want to transmit ti. The antenna gets commanded to point to the selected satellite and the 'channel' selection gets filled with the channel selection stored for this satellite.
2. Select the channel you want to transmit. This applies the polarization / frequency / symbol rate / fec settings stored for this channel to the devices which are configured with this TX-ChannelSelect device.

Instead of selecting the satellite and channel you also may enter the receive parameters at the frequency / pol / symbol rate / fec fields. The TX-Channelselect device forwards these settings to the ancillary devices.

Operation (database edit)

With the second page of the device window of the TX-Channelselect device channel definitions are edited, copied or deleted. All RX/TX-Channelselect devices in a sat-nms M&C share one common channel database. Changes you make at one device apply to all others as well. Selecting channels at his page or changing any channel parameters do not change the receiver settings.

To edit the channel database, first select the satellite which broadcasts the channel you want

to edit. Here a few examples:

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name, this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.
- Save the actual transmit settings as a new channel:
  1. Click 'READ' to fill the fields from actual transmit settings.
  2. Optionally complete the receive frequency / polarization.
  3. Enter a new (for this satellite unique) channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete
  2. Click 'DELETE' All RX/TX-Channelselect devices share the same channel database. Principally you may use any of the devices to edit the database. However, as the TX-ChannelSelect device are bound to one antenna, only channels at satellites which are known to this antenna are accessible with this device. For this reason using an RX-ChannelSelect device for editing channels appears more versatile.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the RX-ChannelSelect device with the IRD and the input switch.

parameter	***description***
polSwitchId	The message / parameter ID used to switch the transmit polarization, e.g. a waveguide switch.
polSwitchHval	The value commanded to the polSwitchId to select horizontal polarization for transmit.
polSwitchVval	The value commanded to the polSwitchId to select vertical polarization for transmit.
antennaName	The name of the Antenna-Pointing device which controls the antenna.
frequencyId	The message / parameter ID used to set the frequency. This parameter sends the true transmit frequency, any frequency converting calculations must be made behind this.
symbolRateId	The message / parameter ID used to set the symbol rate.
fecId	The message / parameter ID used to set the FEC.

parameter	*description*
fault1Id fault2Id fault3Id fault4Id	The device provides four configurable fault inputs. Enter the message ID of the faults to monitor. (e.g. **MOD-1.faults.02** or **ENCODER.fault**) A configured fault inherits the description from the fault it refers to.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.polSwitchId	TEXT	SAVE SETUP	StringRange
config.polSwitchHval	TEXT	SAVE SETUP	StringRange
config.polSwitchVval	TEXT	SAVE SETUP	StringRange
config.antennaName	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.fault1Id	TEXT	SAVE SETUP	StringRange
config.fault2Id	TEXT	SAVE SETUP	StringRange
config.fault3Id	TEXT	SAVE SETUP	StringRange
config.fault4Id	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.lockByIP	CHOICE	SAVE SETUP	EnumRange (NO YES)
antenna	TEXT	R/O	StringRange R/O
satellite	CHOICE		EnumRange (?)
channel	CHOICE		EnumRange (?)
polarization	CHOICE		EnumRange (H V)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
symbolRate	FLOAT		DoubleRange (0.00000 .. 0.00000)
fec	CHOICE		EnumRange (?)
comment	TEXT		StringRange
edit.satellite	CHOICE		EnumRange (?)
edit.channel	CHOICE		EnumRange (?)
edit.rxpol	CHOICE		EnumRange (H V)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (H V)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
edit.fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9 AUTO)
edit.comment	TEXT		StringRange
edit.newchannel	TEXT		StringRange
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
reset	TEXT		StringRange
alias	TEXT	SAVE	StringRange
locked	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
satLocked	TEXT	R/O	StringRange R/O
polLocked	TEXT	R/O	StringRange R/O
togglelock	TEXT	NOPRESET	StringRange
locksource	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fault 01)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Fault 02)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Fault 03)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fault 04)

9.664 TX-LineSettings-CDB

The TX-LineSettings-CDB logical device controls some parameters of an encoder / modulator / power amplifier transmit chain from a common channel database. The device complements the RX-LineSettings-CDB logical device which provides a similar function for receive chains.

TX-LineSettings-CDB supersedes the TX-ChannelSelect-CDB device, which no longer should be used for new applications. The TX-LineSettings-CDB device supports a larger number of parameters, may be used in M&C and in NMS environments and it supports an alternative operation mode / work flow. The name TX-LineSettings-CDB reflects that the set of parameters managed by the device extend the standard channel parameters.

Heart of the TX-LineSettings-CDB device is a central database (CDB) storing the line settings for all satellite channels used in the system. In a NMS environment the database is located in the NMS server, providing an synchronization mechanism which creates local (read-only) copies of the database for offline operation of the VLCs. In a NMS environment, the sat-nms client software looks up the channel lists directly at the NMS database server which significantly reduces the amount of data which has to be exchanged between NMS and VLC. In a M&C environment the CDB is located at the M&C server, no synchronization is required in this case.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The main control (line settings select) page for the device.
- --- The line settings database editor page.
- --- The fault page. There is one fault signal the device (optionally) monitors, this fault is reported as "TX Chain" fault.
- --- The device info page.

- --- The maintenance page. The configuration parameters from this page are described below.

Operation (channel select)

The TX-LineSettings-CDB device supports two different operation modes / work flow settings which are selected by the 'filterBySatellite' configuration parameter.

With 'filterBySatellite=NO' (the default mode) the 'channel' field always shows all line settings data records. The auto-filter mode of the drop down box helps to find the desired satellite channel. Selecting a channel means to set all transmit parameters, the polarization switch and (if configured) also the antenna pointing from the satellite name stored with the selected data record. This operation mode may be used in M&C and NMS environments as well.

With 'filterBySatellite=YES', the 'channel' field only shows the channels of the satellite the antenna actually is pointed to. To move the antenna to another satellite, either the 'satellite' field in the TX-LineSettings-CDB's device screen may me used or the antenne may be controlled via it's own screen. Changing the antenna pointing automatically changes the subset of satellite channels displayed in the 'channels' field. This operation mode is only possible in M&C environments.

The TX-LineSettings-CDB device manages a number of transmit parameters, sets these parameters at the encoder, modulator etc. when a channel is selected. The following parameter may be managed by the TX-LineSettings-CDB device:

- txFrequency
- polarization
- symbolrate / bitrate
- dvb mode
- roll off
- pilots
- modulation
- packet size
- mpeg frame size
- video data rate
- video profile (see below)
- video coding format / profile
- nominal eirp
- reduced eirp for lineup
- audio data rate (1)
- audio data rate (2)
- audio data rate (3)
- audio data rate (4) For an installation of the software not necessarily all of these parameters are available. The TX-LineSettings-CDB device may be configured to ignore some of these parameters.

Please note, that the TX-LineSettings-CDB device neither controls the TX-ON state of the equipment. The operator is responsible to switch off the transmission before applying a

satellite channel data record and also to perform the line up procedure as specified. The LineUp-Control logical device may be used to support the operator with this task. With the LineUp-Control device, the nominal and reduced eirp values from the database are set at this device rather than at the equipment itself. The operator then easily can set the transmit power values required during the line up procedure.

Instead of selecting the satellite and channel you also may enter the transmit parameters at the fields of the device screen. The TX-LineSettings-CDB device forwards these settings to the ancillary devices.

Video profiles

The line settings database contains a parameter called "profile" which controls the basic video encoding mode for the encoder device. This profile contains the MPEG-2 / MPEG-4 selection, the MPEG profile/level and the 4.2.0/4.2.2 color subsampling setting.

Most video encoders don't provide a single parameter to set these values, but use several settings which may differ in name and values they offer to select. To overcome these differences, the TX-LineSettings-CDB device can use special device presets to set the profile parameter in a way individually specialized for the type of video encoder.

To use this feature, store device presets for the profiles you need. Each preset must be named exactly as the profile selection in the channel database (e.g. 'MPEG4-MP@HL-420'). The presets stored for this purpose should be edited in a way, that all parameters which could interfere with other database settings are removed, leaving only the parameters in the preset which set the profile properties.

Then configure the TX-LineSettings-CDB device to route the profile parameter ('profileId' configuration parameter) to the 'applyPreset' parameter of the encoder (e.g. 'ENC.applyPreset'). The TX-LineSettings-CDB device will apply the appropriate device preset when a channel from the database ist selected.

Operation (database edit)

With the second page of the device window of the TX-LineSettings-CDB device channel definitions are edited, copied or deleted. All TX-LineSettings-CDB devices in a sat-nms NMS share one common channel database located at the NMS server. The VLC containing the TX-LineSettings-CDB device requires an online connection to the NMS central database in order to make any changes to the database. Selecting channels at this page or changing any channel parameters do not change the requirement settings.

- Change certain parameters of a channel:
  1. Select the channel to modify
  2. Edit the settings
  3. Click 'SAVE'Please note that if you change the channel name (composed of the satellite / channel id entries), this will store a new channel definition with the modified data, the old channel is kept unchanged in the database.The channel name may consist of letters (a-z, A-Z), digits, space characters and some punctuation characters (,.-_). All other characters are turned to underscore

characters when the channel gets stored.

- Save the actual receive settings as a new channel:
  1. Click 'READ' to fill the fields from actual receive settings.
  2. Optionally complete the transmit parameters for the channel.
  3. Enter the desired satellite / channel name.
  4. Click 'SAVE'
- Delete a channel:
  1. Select the channel to delete
  2. Click 'DELETE'

Local database copy

The following paragraph only applies to NMS installations. M&C installation manage their CDB locally at the M&C server and do not require a local copy of the data.

sat-nms VLCs may be operated locally by connecting an M&C client directly to the VLC. This may be used as a backup operation mode in cases where the NMS server is down or unreachable. To uphold the TX-LineSettings-CDB functionality even in this backup case, the VLC maintains a local copy of the channel database. This copy is used if the CDB at the NMS is unreachable. You cannot edit the channels in this local backup copy, but you can apply stored channels to the IRD.

There are two ways to update the local copy of the channel database at the VLC:

1. At the "Database" page of the device window of a TX-LineSettings-CDB device click "VLC UPD". This requests the complete database contents from the NMS server and stores it locally.
2. Administrators may configure a VLC to update the local database copy in regular intervals. To make a VLC update it's local database copy automatically, add a line **cdb.channels.syncInterval=#.#** to the file vlc.properties and restart the VLC. '#.#' is the time interval in hours.

EBU DREAMS channel list import

The channel database is capable to import channel lists from the EBU DREAMS software. The DREAMS channel lists are distributed as two CSV tables, on for RX and one for TX. In the sat-nms channel database these lists get merged. EBU channels are marked "read-only" in the database, they only may be changed by importing a new EBU channel list.

To import a pair of EBU channel lists, click the "IMPORT" button at the "Database" page of the TX-LineSettings-CDB device's device screen. This opens a new window where the names of the files to import must be entered. The small buttons beside the file name entry fields open a file select dialog which lets you select the file names fom the directory.

After you have set the file names, click "READ". The software reads and parses the CSV files. It also matches corresponding RX and TX channels, because the sat-nms channel database stores only one entry for each channel containing both, the RX and TX parameters.

The software reports the number channels read from each file and the number of channels

matched. Sometimes the numbers slightly differ, e.g. if the original channel names in the RX and TX lists are not really identical. You have to edit and repair the original files if you want to fix this.

If you are satisfied with the parsing result, click "IMPORT". This imports the channel definitions to the sat-nms database. All existing EBU channel definitions are deleted and replaced by the new ones. The user defined channels in the database remain unchanged by this.

Please note, that this procedure copies the EBU channels directly from the client PC to the NMS server, the TX-LineSettings-CDB device is not involved in this and does not recognize the new channels immediately. This happens with the next regular update of the channel list, max 5 minutes later.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the TX-LineSettings-CDB device with the transmit equipment.

parameter	***description***
frequencyId	The message ID to set the transmit frequency.
fecId	The message ID to set the FEC. If the modulator provides a combined modulation/FEC parameter, leave this field empty.
symbolRateId	The message ID to set the symbol rate. Modulators expecting the symbol rate in 'Msps' and in 'sps' are supported. The symbol rate is only set, if the line settings parameter 'useSymbolRate' is set to SYMBOLRATE.
bitRateId	The message ID to set the bit rate. Modulators expecting the bit rate in 'Mbps' and in 'bps' are supported. The bit rate is only set, if the line settings parameter 'useSymbolRate' is set to BITRATE.
dvbModeId	The message ID to set the DVB-S/DVB-S2 mode.
pilotsId	The message ID to switch pilots ON/OFF.
rollOffId	The message ID to set the roll-off parameter.
modulationId	The message ID to set the modulation scheme (e.g. QPSK)
modulationFecId	Some modulator devices provide a combined modulation FEC parameter instead of separate parameters. For such a device enter the message ID of the combined parameter here and leave the modulationId / fecId fields empty.
packetSizeId	The message ID of the modulator packet size parameter (188/204)
frameSizeId	The message ID of the MPEG frame size parameter (short/normal)

parameter	***description***
videoRateId	The message ID to receive the video data rate (Mbit/s). The device sends this parameter only if the line settings data record has the video data rate marked as FIXED.
profileId	The message ID of the video coding profile or the 'applyPreset' parameter of the encoder if the profile shall be set using device presets.
eirpId	The message ID of the parameter to receive the nominal EIRP value for the selected line setting. In most cases this will be the 'nominalEirp' parameter of the LineUp-Control device complementing this device.
reducedEirpId	The message ID of the parameter to receive the reduced (line up) EIRP value for the selected line setting. In most cases this will be the 'reducedEirp' parameter of the LineUp-Control device complementing this device.
audioRateId	The message ID of the encoder audio bitrate parameter (first audio channel).
audio2RateId	The message ID of the encoder audio bitrate parameter (second audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 2 or higher.
audio3RateId	The message ID of the encoder audio bitrate parameter (third audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is 4 or higher.
audio4RateId	The message ID of the encoder audio bitrate parameter (fourth audio channel). The value is only set if the 'audioChannels' parameter in the actual line settings is to 4.
satelliteId	The parameter ID of the 'satellite' parameter of theAntenna-Pointing device of the antenna which is used for transmission.
txFaultId	You may specify the message ID of a fault signal here which is reported as "TX Chain" fault by the device. If you don't want any fault monitoring, leave the field empty.
polarizationId	The message ID to set the transmit polarization. The device sends X / Y from the channel definition to this ID. With a waveguide switch for pol select you may us a Choice-Translator logical device to translate X/Y into the proper switch positions.

parameter	*description*
txLoFrequency	You may set a BUC LO frequency here, e.g. if the transmit equipment is not capable to compute the modulator output frequency by itself. If your equipment is configured to set the transmit frequency in terms of true RF frequency at the antenna, then leave the field empty or set the LO frequency explicitly to 0.
filterBySatellite	Selects the workflow / operation mode of the device. "NO" selects the standard, unfiltered mode: The 'channel' parameter always offers all line settings records stored in the database. "YES" switch to filtered mode, the 'channel' parameter shows only the channel matching the actual satellite addressed by 'satellite'.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.fecId	TEXT	SAVE SETUP	StringRange
config.symbolRateId	TEXT	SAVE SETUP	StringRange
config.bitRateId	TEXT	SAVE SETUP	StringRange
config.dvbModeId	TEXT	SAVE SETUP	StringRange
config.pilotsId	TEXT	SAVE SETUP	StringRange
config.rollOffId	TEXT	SAVE SETUP	StringRange
config.modulationId	TEXT	SAVE SETUP	StringRange
config.modulationFecId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.txFaultId	TEXT	SAVE SETUP	StringRange
config.polarizationId	TEXT	SAVE SETUP	StringRange
config.packetSizeId	TEXT	SAVE SETUP	StringRange
config.frameSizeId	TEXT	SAVE SETUP	StringRange
config.videoRateId	TEXT	SAVE SETUP	StringRange
config.profileId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.reducedEirpId	TEXT	SAVE SETUP	StringRange
config.audioRateId	TEXT	SAVE SETUP	StringRange
config.audio2RateId	TEXT	SAVE SETUP	StringRange
config.audio3RateId	TEXT	SAVE SETUP	StringRange
config.audio4RateId	TEXT	SAVE SETUP	StringRange
config.satelliteId	TEXT	SAVE SETUP	StringRange
config.txLoFrequency	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.filterBySatellite	CHOICE	SAVE SETUP	EnumRange (NO YES)
satellite	TEXT		StringRange DIS
fec	CHOICE		EnumRange DIS (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)

name	*type*	*flags*	*range*
txFrequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
polarization	CHOICE		EnumRange DIS (X Y L R)
comment	TEXT	SAVE	StringRange
symbolRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
bitRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
dvbMode	CHOICE		EnumRange DIS (DVBS DVBS2 NS3 NS4)
rollOff	CHOICE		EnumRange DIS (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
pilots	CHOICE		EnumRange DIS (OFF ON)
modulation	CHOICE		EnumRange DIS (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
packetSize	CHOICE		EnumRange DIS (188 204)
frameSize	CHOICE		EnumRange DIS (SHORT NORMAL)
videoRate	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
profile	CHOICE		EnumRange DIS (UNKNOWN)
eirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
reducedEirp	FLOAT		DoubleRange DIS (0.00 .. 0.00)
audioRate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio2Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio3Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
audio4Rate	CHOICE		EnumRange DIS (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
reloadProfiles	TEXT		StringRange
channel	CHOICE		EnumRange (@@@channels)
reset	TEXT		StringRange

name	*type*	*flags*	*range*
edit.channel	CHOICE		EnumRange (@@@channels)
edit.satellite	TEXT		StringRange
edit.chanid	TEXT		StringRange
edit.rxpol	CHOICE		EnumRange (X Y L R)
edit.rxfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.txpol	CHOICE		EnumRange (X Y L R)
edit.txfreq	FLOAT		DoubleRange (0.000 .. 0.000)
edit.symbolRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.bitRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.useSymbolRate	CHOICE		EnumRange (SYMBOLRATE BITRATE)
edit.fec	CHOICE		EnumRange (AUTO 1/2 1/3 1/4 2/3 3/4 2/5 3/5 4/5 5/6 6/7 7/8 8/9 9/10)
edit.modulation	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16PSK 16QAM 64QAM 256QAM 16APSK 32APSK)
edit.packetSize	CHOICE		EnumRange (188 204)
edit.rollOff	CHOICE		EnumRange (0.02 0.05 0.10 0.15 0.20 0.25 0.35)
edit.dvbMode	CHOICE		EnumRange (DVBS DVBS2 NS3 NS4)
edit.frameSize	CHOICE		EnumRange (SHORT NORMAL)
edit.pilots	CHOICE		EnumRange (OFF ON)
edit.videoRate	FLOAT		DoubleRange (0.0000 .. 0.0000)
edit.videoRateAuto	CHOICE		EnumRange (FIXED AUTO)

name	*type*	*flags*	*range*
edit.profile	CHOICE		EnumRange (MPEG2-MP@ML-420 MPEG2-MP@LL-420 MPEG2-HP@ML-422 MPEG2-HP@LL-422 MPEG2-SP@LL-420 MPEG2-SP@ML-420 MPEG2-MP@H14-420 MPEG2-MP@HL-420 MPEG2-HP@H14-422 MPEG2-HP@HL-422 MPEG2-HP@HL-420 MPEG2-HP@H14-420 MPEG2-HP@ML-420 MPEG4-BP@L2-420 MPEG4-BP@L3-420 MPEG4-BP@L4-420 MPEG4-MP@L3-420 MPEG4-MP@L4-420 MPEG4-MP@L4.2-420 MPEG4-HP@L3-420 MPEG4-HP@L4-420 MPEG4-HP@L4.2-420 MPEG4-HP@L3-422 MPEG4-HP@L4-422 MPEG4-HP@L4.2-422)
edit.eirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.reducedEirp	FLOAT		DoubleRange (0.00 .. 0.00)
edit.autoLineUp	CHOICE		EnumRange DIS (OFF ON)
edit.audioRate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio2Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio3Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audio4Rate	CHOICE		EnumRange (DISABLED 64k 96k 128k 160k 192k 224k 256k 320k 384k)
edit.audioChannels	CHOICE		EnumRange (1 2 3 4)
edit.comment	TEXT		StringRange
edit.user	TEXT		StringRange
edit.brEnable	TEXT	R/O	StringRange R/O
edit.srEnable	TEXT	R/O	StringRange R/O
edit.vrEnable	TEXT	R/O	StringRange R/O
edit.ar2Enable	TEXT	R/O	StringRange R/O
edit.ar3Enable	TEXT	R/O	StringRange R/O
edit.ar4Enable	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
edit.store	TEXT		StringRange
edit.delete	TEXT		StringRange
edit.read	TEXT		StringRange
edit.sync	TEXT		StringRange
display	TEXT	R/O	StringRange R/O
dispColor	TEXT	R/O	StringRange R/O
cdbState	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (TX Chain)

9.665 TX-OFF-Timer

This page describes the device driver and the device window for the *TX-OFF-Timer* device. This device switches off a running transmission at a preprogrammed time within 24 hours. The device also controls an external LCD unit which displays the remaining time of the transmission.

Usage

The timer gets started by entering the desired stop time in the format *hh:mm* . Times in the past are assumed to refer to the next day. While running, the timer sets a flag 'running' which may be used to conditionally enable other M&C functions. When the timer expires, it issues the TX-OFF command defined at the device's maintenance/setup page.

Cisco 2800GSM Interface

The device may be configured to listen to some status variables of a Cisco 2800GSM / VPN router. The state of the VPN tunnel and the GSM receive level (RSSI) value gets displayed in the LCD and the info.gsmstate parameter.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Presents the timer parameters
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
txOffId	The message ID of the parameter controlling the RF ON/OFF state of the transmission, e.g. **EIRP.on** .
txOffCmd	The command / value to be sent to the message id defined above in order to switch off the transmissions. In most cases this will be **OFF** .
txOffDisplay	The text to be shown in the external display while the timer is stopped. The default is **OFF**
showBars	If set to YES, the GSM receive level of the attached Cisco VPN router gets displayed as 0 to 5 bars, similar to a mobile phone display. NO makes the display show the level as digits 0..5.
gsmRssiId	The message ID telling the RSSI level to display, with a Cicso 2800GSM router something like: **CICSO.c3gCurrentGsmRssi** . Expects a level value in dBm to be reported with this parameter.
gsmVpnStatId	The message ID telling the VPN tunnel state. With a Cisco 2800GSM router e.g. **CISCO.rttMonLatestRttOperSense**
gsmRssiTable	This table is used to translate the RSSI level value (dBm) into 0 to 5 bars in the display. The table consists of 5 integer numbers, separated by commas. The first value defines the level necessary to show one bar, the second one is the minimum value for two bars and so on. If you supply less than five values, the device automatically fills the list in 3dB steps.
Serial	Remember to set the line parameters for the serial interface connected to the LCD unit (*9600,N,8,1*).

NMS-VLC clock difference compensation

The *TX-OFF-Timer* takes the clock difference between NMS and VLC into account when calculation the (local) end time of a transmission. The transmission gets stopped at the programmed end time in terms of NMS time. This calculation may interfere with the clock adaption being done by the VLC's clktrim daemon. The latter tunes the VLC clock to run faster or slower about up to 10 minutes/day until the VLC clock matches the NMS time. The clock difference between VLC and NMS by and by becomes smaller. Hence, with large VLC clock deviations, the effective end time may differ from the NMS time. It is recommended not to use the clktrim daemon for VLCs which are only occasionally running for this reason.

Using the TX-OFF-Timer without the LCD unit

The TX-OFF-Timer also may be used without the external LCD unit. To do this, configure the TX-OFF-Timer with a 'null' interface and the 'Logical' protocol in the VLC's device setup. The TX-

OFF-Timer will not occupy a serial interface when configured in this way.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.timeDiff	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.gsmstat	TEXT	R/O	StringRange R/O
config.txOffId	TEXT	SAVE SETUP	StringRange
config.txOffCmd	TEXT	SAVE SETUP	StringRange
config.txOffDisplay	TEXT	SAVE SETUP	StringRange
config.showBars	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.gsmRssiId	TEXT	SAVE SETUP	StringRange
config.gsmVpnStatId	TEXT	SAVE SETUP	StringRange
config.gsmCommFaultId	TEXT	SAVE SETUP	StringRange
config.gsmRssiTable	TEXT	SAVE SETUP	StringRange
txOffTime	TEXT		StringRange
remainingTime	TEXT	R/O	StringRange R/O
running	BOOLEAN	R/O	BooleanRange R/O (true false)
extended	BOOLEAN	R/O	BooleanRange R/O (true false)
extendPossible	BOOLEAN	R/O	BooleanRange R/O (true false)
extend	TEXT		StringRange
gsmstat	TEXT		StringRange

9.666 Tadiran-Scopus-E1000

This page describes the device driver and the device window for the Tadiran Scopus E1000 MPEG Video Encoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- MPEG output parameters.
- --- Video parameters.
- --- Audio parameters.
- --- Auxiliary data parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

There are no configuration parameters for this device driver.

Communication protocol / Interface type

The device driver talks to the RS232 "user data" interface of the encoder using the*Tadiran-Protocol* .

Remarks

- The "user data" RS232 interface is used to control the E1000. Therefore it is not possible to use this interface for low speed data to be encoded into the MPEG stream.
- Not all versions of the E1000 are capable to be controlled via the RS232 interface. Ensure to purchase the E1000 with this option enabled or update the firmware (it's stored on a flash card) later.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
info.codec	TEXT	R/O	StringRange R/O
info.audioChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.srevision	TEXT	R/O	StringRange R/O
info.hrevision	TEXT	R/O	StringRange R/O
info.permcode	TEXT	R/O	StringRange R/O
info.videoifc	TEXT	R/O	StringRange R/O
output.datarate	INTEGER		IntegerRange (1000000 .. 25000000)
output.testmode	CHOICE		EnumRange (OFF AUDIO VIDEO A+V)
output.clksource	CHOICE		EnumRange (INTERN EXTERN)
output.pmtpid	HEX		HexRange (20 .. 1FFF)
output.pcrpid	HEX		HexRange (0 .. 1FFF)
output.camode	CHOICE		EnumRange (NO CA ENCODER CAS-5000 Prop CAS-5000 DVB)
output.profile	CHOICE		EnumRange (4:2:0 4:2:0 LD 4:2:2 4:2:2 LD)
output.dvbtables	CHOICE		EnumRange (OFF ON)
output.packetsize	CHOICE		EnumRange (188 204)
output.program	HEX		HexRange (1 .. 1FFFF)
output.service	TEXT		StringRange
video.source	CHOICE		EnumRange (COMPOSITE S-VHS COMPONENT DIGITAL)
video.format	CHOICE		EnumRange (NTSC PAL)
video.datarate	INTEGER		IntegerRange (1000000 .. 25000000)
video.resolution	CHOICE		EnumRange (720 FULL 704 FULL 640 FULL 544 FULL 368 FULL 352 HALF 352 FULL AUTO)

name	*type*	*flags*	*range*
video.pid	HEX		HexRange (20 .. 1FFF)
video.aspect	CHOICE		EnumRange (4:3 16:9 4:3 AUTO 16:9 AUTO)
video.dnr	CHOICE		EnumRange (OFF LOW MED HIGH)
video.luminance	INTEGER		IntegerRange (0 .. 255)
video.chrominance	INTEGER		IntegerRange (0 .. 255)
video.offset	INTEGER		IntegerRange (0 .. 255)
video.yc-delay	INTEGER		IntegerRange (0 .. 9)
audio.1.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.1.sampling	CHOICE		EnumRange (32 kHz 44.1 kHz 48 kHz)
audio.1.source	CHOICE		EnumRange (EMBEDDED ANALOG DIGITAL AUTO)
audio.1.pid	HEX		HexRange (20 .. 1FFF)
audio.2.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.2.sampling	CHOICE	R/O	EnumRange R/O (32 kHz 44.1 kHz 48 kHz)
audio.2.source	CHOICE		EnumRange (EMBEDDED ANALOG DIGITAL AUTO)
audio.2.pid	HEX		HexRange (20 .. 1FFF)
audio.3.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.3.sampling	CHOICE	R/O	EnumRange R/O (32 kHz 44.1 kHz 48 kHz)
audio.3.source	CHOICE		EnumRange (EMBEDDED ANALOG DIGITAL AUTO)
audio.3.pid	HEX		HexRange (20 .. 1FFF)

name	*type*	*flags*	*range*
audio.4.datarate	CHOICE		EnumRange (OFF 64 kbps 96 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps)
audio.4.sampling	CHOICE	R/O	EnumRange R/O (32 kHz 44.1 kHz 48 kHz)
audio.4.source	CHOICE		EnumRange (EMBEDDED ANALOG DIGITAL AUTO)
audio.4.pid	HEX		HexRange (20 .. 1FFF)
hsdata.datarate	INTEGER		IntegerRange (0 .. 8192000)
hsdata.pid	HEX		HexRange (20 .. 1FFF)
hsdata.clock	CHOICE		EnumRange (64 kHz 512 kHz 1024 kHz 2048 kHz 3072 kHz 4096 kHz 6144 kHz 8192 kHz)
teletext.on	CHOICE		EnumRange (OFF ON)
teletext.pid	HEX		HexRange (20 .. 1FFF)
internal.status	TEXT	R/O	StringRange R/O
internal.ack	TEXT	R/O	StringRange R/O
internal.faults	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Audio codec)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video codec)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video input)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (External clock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Remote access)

9.667 Tandberg-Alteia

This page describes the device driver and the device window for Tandberg Alteia IRDs. The actual version of the driver supports the Alteia / Alteia plus series satellite receivers, not the terrestrial receiver and pure demodulator models.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters.
- --- Contains the parameters concerning the video output (Analog video format selection, output level).
- --- Contains the parameters concerning the audio output (Language selection, L/R/Mono routing, Level).
- --- Contains the MPEG / transport stream routing parameters for the digital output of the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Alteia IRD.

parameter	***description***
address	You must set the device address of the IRD here to make the program talk to the device. The device address is a three digit number which must be entered with leading zeroes. The factory preset address at the IRDs is "000" but may be set to any other value at the IRD frontpanel.
lbandInputs	Select the number of L-band inputs the receiver provides (depending on the model 2 or 4 intputs)
lnbPower	Select if the IRD shall apply LNB power to the actually selected input. **ATTENTION:**
The IRD applies this setting to the input you actually have selected, only. In order to define the LNB power setting for all four inputs, you have to select the input and set the LNB power mode for each of the inputs in turn.	

parameter	***description***
loFreq	Once you enter a value here, the IRD treats the current L-band input as a "single LNB" input and accepts RF input frequencies according to the LO frequency you have entered. **ATTENTION:**
The IRD applies this setting to the input you actually have selected, only. The setting is internally memorized by the IRD and used if this input is selected. In order to set the LO frequencies for all four inputs, you have to select the input and enter the LO frequency for each of the inputs in turn.	
berThreshold	The BER threshold value. The IRD raises an alarm if the bit error rate exceeds this value.
sigThreshold	The signal level threshold value. The IRD raises an alarm if the signal level drops below this value
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the IRD. The proper setting is *9600,N,8,1* .

Operating multiple IRDs at one RS485 interface

The Alteia IRDs use a "multi drop" capable protocol for remote control. If you select a RS485 interface for this, you may control more than one IRD through this interface. To do this:

- Using the IRD's front panel control, set each device to use the RS485 interface.
- For each device at on RS485 bus, select a unique device address. If you operate more than one IRD at a single interface, you should be aware that communication becomes very slow. The computer has to poll a lot of information from each IRD and the line speed is fixed to 9600 baud.

Using dual LNCs and voltage controlled polarization switching

The Alteia IRD is capable to produce different LNB voltages for polarization selection and automatically add a 22 kHz signal for dual LNCs. To make use of these features, configure the IRD as follows:

- At the IRD front panel, configure each input (or at least the input you are going to use) for dual LNB mode, provide the correct LO frequencies and the switch over frequency.
- Enable the LNB power. The IRD now automatically will select the LNB from the RF frequency you enter. It is important not to enter any value to the "loFreq" configuration parameter. This parameter shows the lower LO frequency for the actually selected input, as soon as you change this value, this input will be reset to "single LNB" mode.

To control the polarization by changing the LNB voltage, a hidden parameter calls "polarization" has been implemented with the device driver. If you are using the IRD together with a link management software, you may want to direct the link RX polarization value to this parameter. To change the polarization manually, the fastest way is to enter the low level command at the maintenance page of the user interface: "TUN:POL=HOR" and "TUN:POL=VER" are the quite self-explanatory command you have to enter.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.ca.casid	TEXT	R/O	StringRange R/O
info.ca.codeversion	TEXT	R/O	StringRange R/O
info.ca.bootversion	TEXT	R/O	StringRange R/O
info.ca.modelno	TEXT	R/O	StringRange R/O
info.ca.hardware	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.ca.manufacturer	TEXT	R/O	StringRange R/O
info.ca.download	TEXT	R/O	StringRange R/O
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (4 2)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON. BST)
config.loFreq	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.berThreshold	TEXT	SETUP	StringRange
config.sigThreshold	INTEGER	SETUP	IntegerRange (0 .. 255)
config.errFrame	CHOICE	SETUP	EnumRange (FREEZE BLACK)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.numServices	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.numAudio	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ack	TEXT	R/O	StringRange R/O
internal.flags	HEX	R/O	HexRange R/O (0 .. 0)
internal.scnt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input	CHOICE		EnumRange (1 2 3 4)
polarization	CHOICE		EnumRange (HOR VER)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE		EnumRange (BPSK QPSK 8PSK)
fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 6/7 7/8 8/9)
programNo	INTEGER		IntegerRange (0 .. 99)
programList	TEXT	R/O	StringRange R/O
audioList	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
actualProgram	TEXT	R/O	StringRange R/O
video.fmt625	CHOICE		EnumRange (PALI PALB PALN)
video.fmt525	CHOICE		EnumRange (NTSC NTSN PALM)
video.level	INTEGER		IntegerRange (-30 .. 30)
video.test	CHOICE		EnumRange (NRM 625.1 625.2 625.3 625.4 625.5 625.6 625.7 625.8 625.9 625.10 625.11 625.12 625.13 625.14 625.15 625.16 525.1 525.2 525.3 525.4 525.5 525.6 525.7 525.8 525.9 525.10 525.11 525.12 525.13 525.14 525.15 525.16)
audio.1.program	INTEGER		IntegerRange (0 .. 0)
audio.1.routing	CHOICE		EnumRange (NRM MON LFT RGT)
audio.1.output	CHOICE		EnumRange (ANALOG AES/EBU SPDIF AC3)
audio.1.level	INTEGER		IntegerRange (6 .. 18)
audio.1.language	TEXT		StringRange
audio.1.test	CHOICE		EnumRange (NRM TEST-1 TEST-2 TEST-3 TEST-4 TEST-5)
audio.1.info	TEXT	R/O	StringRange R/O
audio.2.program	INTEGER		IntegerRange (0 .. 0)
audio.2.routing	CHOICE		EnumRange (NRM MON LFT RGT)
audio.2.output	CHOICE		EnumRange (ANALOG AES/EBU SPDIF AC3)
audio.2.level	INTEGER		IntegerRange (6 .. 18)
audio.2.language	TEXT		StringRange
audio.2.test	CHOICE		EnumRange (NRM TEST-1 TEST-2 TEST-3 TEST-4 TEST-5)
audio.2.info	TEXT	R/O	StringRange R/O
tsout.routing	CHOICE		EnumRange (PRE-CA POST-CA)
tsout.fibre	CHOICE		EnumRange (ON OFF)

name	*type*	*flags*	*range*
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ca	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ber	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.signal	INTEGER	R/O	IntegerRange R/O (0 .. 255)
state.aspect	TEXT	R/O	StringRange R/O
state.lines	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O
ca.status	TEXT	R/O	StringRange R/O
ca.service	TEXT	R/O	StringRange R/O
ca.rasmode	CHOICE		EnumRange (DISABLED FIXED DSNG SEC-CA)
ca.bissmode	CHOICE		EnumRange (DISABLED MODE 1 MODE E)
ca.bisskey1	TEXT		StringRange
ca.bisskey2	TEXT		StringRange
ca.bissbits	TEXT		StringRange
ca.dsngkey	TEXT		StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Signal level)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High BER)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Demod Lock)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Conditional Access)

9.668 Tandberg-E57XX-SNMP

This page describes the device driver and the device window for the Tandberg E57XX MPEG video encoder family.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- MPEG output parameters
- --- C/A encryption parameters
- --- Video parameters
- --- Audio 1 parameters
- --- Audio 2 parameters
- --- VBI / data parameters
- --- Device fault list
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

parameter	*description*
address	Enter the IP address of the encoder here, followed by the community string 'private'. Example:
10.0.0.1 private	
hsyncTerm	Switches the termination of the HSYNC input
audio1Term	Switches the termination of the AUDIO 1 input
audio2Term	Switches the termination of the AUDIO 2 input

parameter	*description*
actionOnPidErr	With AUTO-CORRECT, the encoder reassigns PIDs if used more than once. With RAISE-FAULT the encoder states a fault in this case an produces no output stream.
freqMode	Configures if the modulator output frequency is set in terms of IF frequency (70 MHz or L-band) or in term of RF frequency. The latter setting only makes sense with an encoder equipped with L-band modulator and BUC. The BUC LO frequency must be set at the encoder's front panel.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrelease	TEXT	R/O	StringRange R/O
info.fwrelease	TEXT	R/O	StringRange R/O
info.hwrelease	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.modules	TEXT	R/O	StringRange R/O
info.hdVideo	TEXT	R/O	StringRange R/O
info.modulator	TEXT	R/O	StringRange R/O
info.dvbs2	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.audioModules	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.hsyncTerm	CHOICE	SETUP	EnumRange (75 Ohms HI-Z)
config.audio1Term	CHOICE	SETUP	EnumRange (20 kOhms 600 Ohms)
config.audio2Term	CHOICE	SETUP	EnumRange (20 kOhms 600 Ohms)
config.actionOnPidErr	CHOICE	SETUP	EnumRange (AUTO-CORRECT RAISE-FAULT)
config.freqMode	CHOICE	SAVE SETUP	EnumRange (IF RF)
internal.sysHealth	HEX	R/O	HexRange R/O (0 .. 0)
internal.muxHealth	HEX	R/O	HexRange R/O (0 .. 0)
internal.vidHealth	HEX	R/O	HexRange R/O (0 .. 0)
internal.aud1Health	HEX	R/O	HexRange R/O (0 .. 0)
internal.aud2Health	HEX	R/O	HexRange R/O (0 .. 0)
internal.modHealth	HEX	R/O	HexRange R/O (0 .. 0)
internal.rs422Health	HEX	R/O	HexRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
faultList	TEXT	R/O	StringRange R/O
output.mux	CHOICE		EnumRange (OFF ON)
output.dest	CHOICE		EnumRange (ASI SATMOD)
output.datarate	FLOAT		DoubleRange (0.00000 .. 0.00000)
output.format	CHOICE		EnumRange (BASEBAND IF-OUTPUT)
output.program	INTEGER		IntegerRange (1 .. 65535)
output.netid	INTEGER		IntegerRange (0 .. 65535)
output.tsid	INTEGER		IntegerRange (0 .. 65535)

name	*type*	*flags*	*range*
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.network	TEXT		StringRange
output.pmtpid	INTEGER		IntegerRange (0 .. 8190)
output.pcrpid	INTEGER		IntegerRange (0 .. 8190)
output.clksource	CHOICE		EnumRange (LOCAL VIDEO HSYNC)
output.serviceLevel	CHOICE		EnumRange (ON RESERVE OFF)
output.serviceType	CHOICE		EnumRange (TV RADIO TELETEXT DATA)
output.streamIdDesc	CHOICE		EnumRange (OFF ON)
output.dolbyAc3Desc	CHOICE		EnumRange (DVB&ATSC DVB-ONLY ATSC-ONLY)
mod.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
mod.rfon	CHOICE		EnumRange (OFF ON)
mod.power	FLOAT		DoubleRange (-20.0 .. 5.0)
mod.standard	CHOICE		EnumRange (DVB-S DVB-S2)
mod.modulation	CHOICE		EnumRange (OFF ON)
mod.type	CHOICE		EnumRange (QPSK 8PSK)
mod.fec	CHOICE		EnumRange (1/1 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 7/8 8/9 9/10)
mod.symbolrate	FLOAT		DoubleRange (0.00000 .. 0.00000)
mod.spectrum	CHOICE		EnumRange (NORM INVERT)
mod.rollOff	CHOICE		EnumRange (20 % 25 % 30 % 35 %)
mod.format	CHOICE		EnumRange (BASEBAND IF-OUTPUT)
mod.pilot	CHOICE		EnumRange (OFF ON)
mod.mapMode	CHOICE		EnumRange (PEAK PWR MEAN PWR)
mod.frameSize	CHOICE		EnumRange (NORMAL SHORT)

name	*type*	*flags*	*range*
mod.goldSeqNum	INTEGER		IntegerRange (0 .. 262141)
mod.ifTilt	FLOAT		DoubleRange (-0.040 .. 0.040)
ca.type	CHOICE		EnumRange (OFF RAS-1 RAS-2 BISS-1 BISS-E BISS-2)
ca.scrambling	CHOICE		EnumRange (OFF ON FIXED-1 FIXED-2 FIXED-3)
ca.rasKey	INTEGER		IntegerRange (0 .. 0)
ca.injected	TEXT		StringRange
ca.clearsw	TEXT		StringRange
ca.cryptedsw	TEXT		StringRange
video.source	CHOICE		EnumRange (OFF SDI TEST-1 TEST-2 TEST-3 TEST-4 TEST-5 BLACK PAL-I PAL-M NTSC-PED NTSC-NOP HD-SDI HD-TEST HD-BLACK)
video.frameRate	CHOICE		EnumRange (23.98 Hz 24 Hz 25 Hz 29.97 Hz 30 Hz 50 Hz 59.94 Hz 60 Hz)
video.loss	CHOICE		EnumRange (FREEZE BLACK BARS FREEZE+TXT BLACK+TXT BARS+TXT OSD NO-PID NO-OUTPUT)
video.maxStillPeriod	INTEGER		IntegerRange (0 .. 255)
video.adaptPreproc	CHOICE		EnumRange (OFF ON)
video.hdnr	CHOICE		EnumRange (OFF LEVEL-1 LEVEL-2 LEVEL-3 LEVEL-4)
video.borderProc.t	INTEGER		IntegerRange (0 .. 100)
video.borderProc.b	INTEGER		IntegerRange (0 .. 100)
video.borderProc.l	INTEGER		IntegerRange (0 .. 100)
video.borderProc.r	INTEGER		IntegerRange (0 .. 100)
video.profile	CHOICE		EnumRange (MP@ML 422P@ML MP@HL 422P@HL)

name	*type*	*flags*	*range*
video.compression	CHOICE		EnumRange (SEAMLESS-1 SEAMLESS-2 SEAMLESS-3 SEAMLESS-4 SEAMLESS-5 SEAMLESS-6 STANDARD LOW-DELAY VERY-LOW-DELAY MEGA-LOW-DELAY SIF-LOW-BITRATE SIF-LOW-DELAY)
video.datarate	INTEGER		IntegerRange (0 .. 0)
video.hresolution	INTEGER		IntegerRange (0 .. 0)
video.vresolution	INTEGER		IntegerRange (0 .. 0)
video.aspect	CHOICE		EnumRange (1:1 4:3 16:9 2.21:1)
video.gopStruct	INTEGER		IntegerRange (0 .. 0)
video.gopLength	INTEGER		IntegerRange (1 .. 180)
video.adaptiveGop	CHOICE		EnumRange (OFF ON)
video.32pulldown	CHOICE		EnumRange (OFF CHIPSET PREPROC)
video.pid	INTEGER		IntegerRange (0 .. 8191)
video.tag	INTEGER		IntegerRange (0 .. 255)
video.sceneCutDet	CHOICE		EnumRange (OFF ON)
video.pesHeader	CHOICE		EnumRange (PER-GOP PER-PICT)
video.intraDcPrec	CHOICE		EnumRange (AUTO 8 BITS 9 BITS 10 BITS)
video.rateBuffer	CHOICE		EnumRange (SMALL MEDIUM LARGE)
video.delay	INTEGER		IntegerRange (0 .. 0)
video.fieldFrame	CHOICE		EnumRange (AUTO FIELD FRAME)
video.VBVdelay	CHOICE		EnumRange (OFF ON)
video.AFDlocation	CHOICE		EnumRange (SEQUENCE PICTURE)
video.embeddedSource	CHOICE		EnumRange (SD-SDI HD-SDI)

name	*type*	*flags*	*range*
audio.1.source	CHOICE		EnumRange (OFF MUTE TEST TONE ANALOG DIGITAL SDI-EMBEDDED-1 SDI-EMBEDDED-2 SDI-EMBEDDED-3 SDI-EMBEDDED-4 DIGITAL-VIA-SRC DIGITAL-2-VIA-SRC EMBEDDED-1-VIA-SRC EMBEDDED-2-VIA-SRC EMBEDDED-3-VIA-SRC EMBEDDED-4-VIA-SRC XLR-MODULE HD-AVC-EMBEDDED-GRP-1-2 HD-AVC-EMBEDDED-GRP-3-4 HD-AVC-EMBEDDED-1 HD-AVC-EMBEDDED-2 HD-AVC-EMBEDDED-3 HD-AVC-EMBEDDED-4 HD-AVC-EMBEDDED-5 HD-AVC-EMBEDDED-6 HD-AVC-EMBEDDED-7 HD-AVC-EMBEDDED-8)
audio.1.coding	CHOICE		EnumRange (DOLBY-AC3 LINEAR-PCM MPEG-2 AC3-PASSTHRU DOLBY-E-PASSTHRU DTS-PASSTHRU PCM-VIA-SRC)
audio.1.alignment	CHOICE		EnumRange (UNALIGNED PES-HDR)
audio.1.gain	CHOICE		EnumRange (12 dB 18 dB 15 dB 16 dB 21 dB 22 dB 24 dB)
audio.1.datarate	CHOICE		EnumRange (32 kbps 40 kbps 48 kbps 56 kbps 64 kbps 80 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps 448 kbps 512 kbps 576 kbps 640 kbps)
audio.1.mode	CHOICE		EnumRange (STEREO J-STEREO DUAL-MONO MONO-LEFT ONE-PLUS-ONE C-LEFT L-R L-C-R L-R-S L-C-R-S L-R-LS-RS L-C-R-LS-RS AUDIO-DESCR-SERV MONO-RIGHT C-RIGHT)
audio.1.samplingRate	CHOICE	R/O	EnumRange R/O (48 kHz 44.1 kHz 32 kHz)
audio.1.vpsCtrlMode	CHOICE		EnumRange (OFF ON)
audio.1.delay	INTEGER		IntegerRange (0 .. 0)
audio.1.lipSync	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
audio.1.useMinDelay	CHOICE		EnumRange (OFF ON)
audio.1.lipSyncOffset	INTEGER		IntegerRange (0 .. 0)
audio.1.lang	TEXT		StringRange
audio.1.copyright	CHOICE		EnumRange (OFF ON)
audio.1.original	CHOICE		EnumRange (OFF ON)
audio.1.pid	INTEGER		IntegerRange (0 .. 8191)
audio.1.tag	INTEGER		IntegerRange (0 .. 255)
audio.1.pcrOnPid	CHOICE		EnumRange (OFF ON)
audio.1.opDigitalLoss	CHOICE		EnumRange (SILENCE NO-PID NO-ASI)
audio.2.source	CHOICE		EnumRange (OFF MUTE TEST TONE ANALOG DIGITAL SDI-EMBEDDED-1 SDI-EMBEDDED-2 SDI-EMBEDDED-3 SDI-EMBEDDED-4 DIGITAL-VIA-SRC DIGITAL-2-VIA-SRC EMBEDDED-1-VIA-SRC EMBEDDED-2-VIA-SRC EMBEDDED-3-VIA-SRC EMBEDDED-4-VIA-SRC XLR-MODULE HD-AVC-EMBEDDED-GRP-1-2 HD-AVC-EMBEDDED-GRP-3-4 HD-AVC-EMBEDDED-1 HD-AVC-EMBEDDED-2 HD-AVC-EMBEDDED-3 HD-AVC-EMBEDDED-4 HD-AVC-EMBEDDED-5 HD-AVC-EMBEDDED-6 HD-AVC-EMBEDDED-7 HD-AVC-EMBEDDED-8)
audio.2.coding	CHOICE		EnumRange (DOLBY-AC3 LINEAR-PCM MPEG-2 AC3-PASSTHRU DOLBY-E-PASSTHRU DTS-PASSTHRU PCM-VIA-SRC)
audio.2.alignment	CHOICE		EnumRange (UNALIGNED PES-HDR)
audio.2.gain	CHOICE		EnumRange (12 dB 18 dB 15 dB 16 dB 21 dB 22 dB 24 dB)

name	*type*	*flags*	*range*
audio.2.datarate	CHOICE		EnumRange (32 kbps 40 kbps 48 kbps 56 kbps 64 kbps 80 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps 448 kbps 512 kbps 576 kbps 640 kbps)
audio.2.mode	CHOICE		EnumRange (STEREO J-STEREO DUAL-MONO MONO-LEFT ONE-PLUS-ONE C-LEFT L-R L-C-R L-R-S L-C-R-S L-R-LS-RS L-C-R-LS-RS AUDIO-DESCR-SERV MONO-RIGHT C-RIGHT)
audio.2.samplingRate	CHOICE	R/O	EnumRange R/O (48 kHz 44.1 kHz 32 kHz)
audio.2.vpsCtrlMode	CHOICE		EnumRange (OFF ON)
audio.2.delay	INTEGER		IntegerRange (0 .. 0)
audio.2.lipSync	CHOICE		EnumRange (OFF ON)
audio.2.useMinDelay	CHOICE		EnumRange (OFF ON)
audio.2.lipSyncOffset	INTEGER		IntegerRange (0 .. 0)
audio.2.lang	TEXT		StringRange
audio.2.copyright	CHOICE		EnumRange (OFF ON)
audio.2.original	CHOICE		EnumRange (OFF ON)
audio.2.pid	INTEGER		IntegerRange (0 .. 8191)
audio.2.tag	INTEGER		IntegerRange (0 .. 255)
audio.2.pcrOnPid	CHOICE		EnumRange (OFF ON)
audio.2.opDigitalLoss	CHOICE		EnumRange (SILENCE NO-PID NO-ASI)

name	*type*	*flags*	*range*
audio.3.source	CHOICE		EnumRange (OFF MUTE TEST TONE ANALOG DIGITAL SDI-EMBEDDED-1 SDI-EMBEDDED-2 SDI-EMBEDDED-3 SDI-EMBEDDED-4 DIGITAL-VIA-SRC DIGITAL-2-VIA-SRC EMBEDDED-1-VIA-SRC EMBEDDED-2-VIA-SRC EMBEDDED-3-VIA-SRC EMBEDDED-4-VIA-SRC XLR-MODULE HD-AVC-EMBEDDED-GRP-1-2 HD-AVC-EMBEDDED-GRP-3-4 HD-AVC-EMBEDDED-1 HD-AVC-EMBEDDED-2 HD-AVC-EMBEDDED-3 HD-AVC-EMBEDDED-4 HD-AVC-EMBEDDED-5 HD-AVC-EMBEDDED-6 HD-AVC-EMBEDDED-7 HD-AVC-EMBEDDED-8)
audio.3.coding	CHOICE		EnumRange (DOLBY-AC3 LINEAR-PCM MPEG-2 AC3-PASSTHRU DOLBY-E-PASSTHRU DTS-PASSTHRU PCM-VIA-SRC)
audio.3.alignment	CHOICE		EnumRange (UNALIGNED PES-HDR)
audio.3.gain	CHOICE		EnumRange (12 dB 18 dB 15 dB 16 dB 21 dB 22 dB 24 dB)
audio.3.datarate	CHOICE		EnumRange (32 kbps 40 kbps 48 kbps 56 kbps 64 kbps 80 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps 448 kbps 512 kbps 576 kbps 640 kbps)
audio.3.mode	CHOICE		EnumRange (STEREO J-STEREO DUAL-MONO MONO-LEFT ONE-PLUS-ONE C-LEFT L-R L-C-R L-R-S L-C-R-S L-R-LS-RS L-C-R-LS-RS AUDIO-DESCR-SERV MONO-RIGHT C-RIGHT)
audio.3.samplingRate	CHOICE	R/O	EnumRange R/O (48 kHz 44.1 kHz 32 kHz)
audio.3.vpsCtrlMode	CHOICE		EnumRange (OFF ON)
audio.3.delay	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
audio.3.lipSync	CHOICE		EnumRange (OFF ON)
audio.3.useMinDelay	CHOICE		EnumRange (OFF ON)
audio.3.lipSyncOffset	INTEGER		IntegerRange (0 .. 0)
audio.3.lang	TEXT		StringRange
audio.3.copyright	CHOICE		EnumRange (OFF ON)
audio.3.original	CHOICE		EnumRange (OFF ON)
audio.3.pid	INTEGER		IntegerRange (0 .. 8191)
audio.3.tag	INTEGER		IntegerRange (0 .. 255)
audio.3.pcrOnPid	CHOICE		EnumRange (OFF ON)
audio.3.opDigitalLoss	CHOICE		EnumRange (SILENCE NO-PID NO-ASI)
audio.4.source	CHOICE		EnumRange (OFF MUTE TEST TONE ANALOG DIGITAL SDI-EMBEDDED-1 SDI-EMBEDDED-2 SDI-EMBEDDED-3 SDI-EMBEDDED-4 DIGITAL-VIA-SRC DIGITAL-2-VIA-SRC EMBEDDED-1-VIA-SRC EMBEDDED-2-VIA-SRC EMBEDDED-3-VIA-SRC EMBEDDED-4-VIA-SRC XLR-MODULE HD-AVC-EMBEDDED-GRP-1-2 HD-AVC-EMBEDDED-GRP-3-4 HD-AVC-EMBEDDED-1 HD-AVC-EMBEDDED-2 HD-AVC-EMBEDDED-3 HD-AVC-EMBEDDED-4 HD-AVC-EMBEDDED-5 HD-AVC-EMBEDDED-6 HD-AVC-EMBEDDED-7 HD-AVC-EMBEDDED-8)
audio.4.coding	CHOICE		EnumRange (DOLBY-AC3 LINEAR-PCM MPEG-2 AC3-PASSTHRU DOLBY-E-PASSTHRU DTS-PASSTHRU PCM-VIA-SRC)
audio.4.alignment	CHOICE		EnumRange (UNALIGNED PES-HDR)
audio.4.gain	CHOICE		EnumRange (12 dB 18 dB 15 dB 16 dB 21 dB 22 dB 24 dB)

name	*type*	*flags*	*range*
audio.4.datarate	CHOICE		EnumRange (32 kbps 40 kbps 48 kbps 56 kbps 64 kbps 80 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 224 kbps 256 kbps 320 kbps 384 kbps 448 kbps 512 kbps 576 kbps 640 kbps)
audio.4.mode	CHOICE		EnumRange (STEREO J-STEREO DUAL-MONO MONO-LEFT ONE-PLUS-ONE C-LEFT L-R L-C-R L-R-S L-C-R-S L-R-LS-RS L-C-R-LS-RS AUDIO-DESCR-SERV MONO-RIGHT C-RIGHT)
audio.4.samplingRate	CHOICE	R/O	EnumRange R/O (48 kHz 44.1 kHz 32 kHz)
audio.4.vpsCtrlMode	CHOICE		EnumRange (OFF ON)
audio.4.delay	INTEGER		IntegerRange (0 .. 0)
audio.4.lipSync	CHOICE		EnumRange (OFF ON)
audio.4.useMinDelay	CHOICE		EnumRange (OFF ON)
audio.4.lipSyncOffset	INTEGER		IntegerRange (0 .. 0)
audio.4.lang	TEXT		StringRange
audio.4.copyright	CHOICE		EnumRange (OFF ON)
audio.4.original	CHOICE		EnumRange (OFF ON)
audio.4.pid	INTEGER		IntegerRange (0 .. 8191)
audio.4.tag	INTEGER		IntegerRange (0 .. 255)
audio.4.pcrOnPid	CHOICE		EnumRange (OFF ON)
audio.4.opDigitalLoss	CHOICE		EnumRange (SILENCE NO-PID NO-ASI)
teletext.enable	CHOICE		EnumRange (OFF ON)
teletext.pid	INTEGER		IntegerRange (0 .. 8191)
teletext.tag	INTEGER		IntegerRange (0 .. 255)
rs232.enable	CHOICE		EnumRange (OFF ON-TANDBERG ON-DVB ON-MOTOROLA ON-WEGENER)

name	*type*	*flags*	*range*
rs232.pid	INTEGER		IntegerRange (0 .. 8191)
rs232.tag	INTEGER		IntegerRange (0 .. 255)
rs232.datarate	CHOICE		EnumRange (1200 bps 2400 bps 4800 bps 9600 bps 19200 bps 38400 bps)
rs422.enable	CHOICE		EnumRange (OFF ON)
rs422.pid	INTEGER		IntegerRange (0 .. 8191)
rs422.tag	INTEGER		IntegerRange (0 .. 255)
rs422.datarate	CHOICE		EnumRange (64 kbps 128 kbps 192 kbps 256 kbps 320 kbps 384 kbps 448 kbps 512 kbps 576 kbps 640 kbps 704 kbps 768 kbps 832 kbps 896 kbps 960 kbps 1024 kbps 1088 kbps 1152 kbps 1216 kbps 1280 kbps 1344 kbps 1408 kbps 1472 kbps 1536 kbps 1600 kbps 1664 kbps 1728 kbps 1792 kbps 1856 kbps 1920 kbps 1984 kbps 2048 kbps 56 kbps 112 kbps 168 kbps 224 kbps 280 kbps 336 kbps 392 kbps 448 kbps 504 kbps 560 kbps 616 kbps 672 kbps 728 kbps 784 kbps 840 kbps 896 kbps 952 kbps 1008 kbps 1064 kbps 1120 kbps 1176 kbps 1232 kbps 1288 kbps 1344 kbps 1400 kbps)
rs422.testMode	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (System)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Modulator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Multiplexer)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Audio 1)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Audio 2)
faults.07	ALARM	R/O	AlarmFlagRange R/O (RS422 Data)

name	*type*	*flags*	*range*
faults.08	ALARM	R/O	AlarmFlagRange R/O (Modulator clock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Modulator input)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Video input lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Audio 1 input)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Audio 2 input)
faults.88	ALARM	R/O	AlarmFlagRange R/O (E5XX Summary)

9.669 Tandberg-RX1290-SNMP

This page describes the device driver and the device window for Tandberg RX1290 DVB-S2 / HDTV IRD, controlled via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video, video test output and embedded audio assignment.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream.
- --- Contains the parameters concerning the ASI / SDI output ports of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the RX1290.

parameter	*description*

parameter	***description***
address	Enter the IP address of the IRD here, in 'dotted quad' notation. The IRD requires the community string 'private' to be appended to the address. Example **10.0.0.1 private**
lbandInputs	This parameter should be set to the number of L-band input the IRD is equipped with (1/2/4).
22kHz	Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Selects if the IRD shall apply LNB power to the input.
lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
lnbPolSwitch	If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
lnbBoost	Raises the LNB supply voltage by 1V if set to ON (long cable compensation).
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.
spectrumSense	Controls the spectrum inversion setting of the IRD. AUTO should work in most cases (switches to NORMAL for Ku-band LO Frequencies, INVERTED for C-band frequencies).
serviceHunt	If set to ON, the IRD receives the first available program in a stream when tuned to a new signal. With OFF, the IRD insists in receiving the service ID / PIDs which were set before. For SNG hub applications you will usually select ON, for distribution / relaying applications OFF.

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.setDelay	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 3.0)
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (1 2 4)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.spectrumSense	CHOICE	SETUP	EnumRange (AUTO NORMAL INVERTED)
config.serviceHunt	CHOICE	SETUP	EnumRange (OFF ON)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.ix	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dvbs2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input	CHOICE		EnumRange (1 2 3 4)
dvbmode	CHOICE		EnumRange DIS (DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	CHOICE	R/O	EnumRange R/O (AUTO)
fec	CHOICE	R/O	EnumRange R/O (NONE 1/2 2/3 3/4 5/6 6/7 7/8 8/9 AUTO 1/4 1/3 2/5 3/5 4/5 9/10)
searchRange	INTEGER		IntegerRange (0 .. 5000)
polarization	CHOICE		EnumRange (X Y)
rolloff	CHOICE		EnumRange DIS (35% 25% 20%)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
actualProgram	TEXT	R/O	StringRange R/O
output.tsOutput	CHOICE		EnumRange (DISABLED POST INPUT POST TS DESCRAMBLE POST ES DESCRAMBLE)
output.conn1	CHOICE		EnumRange (DISABLED ASI SD-SDI HD-SDI AUTO)
output.conn2	CHOICE		EnumRange (DISABLED ASI SD-SDI HD-SDI AUTO)
output.conn3	CHOICE		EnumRange (DISABLED ASI SD-SDI HD-SDI AUTO)
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
video.pcrpid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
video.test	CHOICE		EnumRange (OFF BARS MOVINGBAR BORDER FULLBARS PATHOLOGICAL MULTIBURST LINEUP CONTRAST)
video.testStandard	CHOICE		EnumRange (AUTO 1080i 30Hz 1080i 29.97Hz 1080i 25Hz 1080i 24Hz 1080i 23.976Hz 720p 60Hz 720p 59.94Hz 720p 50Hz 480i 29.97Hz 576i 25Hz)
video.input	CHOICE		EnumRange (ASI SAT)
video.fail	CHOICE		EnumRange (FREEZE BLACK NO SYNC BLUE BARS BLUE+TXT RED+TXT)
video.framesync	CHOICE		EnumRange (OFF ON)
video.mode	CHOICE		EnumRange (PAL-625 NTSC-525)
video.625mode	CHOICE		EnumRange (PAL-BGI PAL-N PAL-NCMB)
video.525mode	CHOICE		EnumRange (NTSC-M PAL-M PAL-525)
video.aspect	CHOICE		EnumRange (4/3 16/9)
video.trisync	CHOICE		EnumRange (ON ALL ON G/Y NONE)
video.rgbOutput	CHOICE		EnumRange (RGB YPbPr)
video.embeddedAudio1	CHOICE		EnumRange (GROUP1 GROUP2 GROUP3 GROUP4 NONE)

name	*type*	*flags*	*range*
video.embeddedAudio2	CHOICE		EnumRange (GROUP1 GROUP2 GROUP3 GROUP4 NONE)
video.embeddedAudio3	CHOICE		EnumRange (GROUP1 GROUP2 GROUP3 GROUP4 NONE)
video.embeddedAudio4	CHOICE		EnumRange (GROUP1 GROUP2 GROUP3 GROUP4 NONE)
video.standard	TEXT	R/O	StringRange R/O
video.colorType	TEXT	R/O	StringRange R/O
audio.1.program	INTEGER		IntegerRange (0 .. 8191)
audio.1.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.1.output	CHOICE		EnumRange (AES AC3)
audio.1.level	INTEGER		IntegerRange (12 .. 24)
audio.1.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.1.delay	INTEGER		IntegerRange (-49 .. 49)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.program	INTEGER		IntegerRange (0 .. 8191)
audio.2.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.2.output	CHOICE		EnumRange (AES AC3)
audio.2.level	INTEGER		IntegerRange (12 .. 24)
audio.2.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.2.delay	INTEGER		IntegerRange (-49 .. 49)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.program	INTEGER		IntegerRange (0 .. 8191)
audio.3.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.3.output	CHOICE		EnumRange (AES AC3)
audio.3.level	INTEGER		IntegerRange (12 .. 24)

name	*type*	*flags*	*range*
audio.3.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.3.delay	INTEGER		IntegerRange (-49 .. 49)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.program	INTEGER		IntegerRange (0 .. 8191)
audio.4.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.4.output	CHOICE		EnumRange (AES AC3)
audio.4.level	INTEGER		IntegerRange (12 .. 24)
audio.4.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.4.delay	INTEGER		IntegerRange (-49 .. 49)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.plen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ca.bissmode	CHOICE		EnumRange (BISS-1 E-FIXED E-TTV E-USER1 E-USER2)
ca.rasmode	CHOICE		EnumRange (FIXED DSNG)
ca.bisskey	TEXT		StringRange
ca.raskey	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.670 Tandberg-RX8200-SNMP

This page describes the device driver and the device window for Tandberg/Ericsson/Mediakind RX8200 DVB-S2 / HDTV IRD, controlled via SNMP.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video decoding end video test output.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream. Unlike with other IRD drivers this page also controls the SDI embedded audio assignments.
- --- Contains the parameters concerning the ASI / SDI output ports of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the RX8200.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- lnbVoltage --- Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdge --- The frequency where the software switches the LO frequencies.
- spectrumSense --- Controls the spectrum inversion setting of the IRD. AUTO should work in most cases (switches to NORMAL for Ku-band LO Frequencies, INVERTED for C-band frequencies).
- serviceHunt --- If set to ON, the IRD receives the first available program in a stream when tuned to a new signal. With OFF, the IRD insists in receiving the service ID / PIDs which were set before. For SNG hub applications you will usually select ON, for distribution / relaying applications OFF.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The RX8200 provides wide functions to monitor the state and control the assignment of SDI embedded audio streams. To present these functions clearly layed out, the embedded audio parameters have been moved to the audio page with this IRD.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
info.lbandInputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.audioChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.swVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.spectrumSense	CHOICE	SETUP	EnumRange (AUTO NORMAL INVERTED)
config.serviceHunt	CHOICE	SETUP	EnumRange (OFF ON)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)
config.hasMultiStream	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasIpInput	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.ix	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dvbs2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.card	TEXT	R/O	StringRange R/O
internal.ipcard	TEXT	R/O	StringRange R/O
internal.state.ebno	TEXT	R/O	StringRange R/O
internal.state.margin	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
input	CHOICE		EnumRange (1 2 3 4 5 6 7 8)

name	*type*	*flags*	*range*
dvbmode	CHOICE		EnumRange (DVBS DVBS2 AUTO UNKNOWN)
modCod	CHOICE		EnumRange DIS (AUTO DVBS-QPSK-1/2 DVBS-QPSK-2/3 DVBS-QPSK-3/4 DVBS-QPSK-5/6 DVBS-QPSK-6/7 DVBS-QPSK-7/8 DVBS-8PSK-2/3 DVBS-8PSK-5/6 DVBS-8PSK-8/9 DVBS-8PSK-16QAM-3/4 DVBS-8PSK-16QAM-7/8 DVBS2-QPSK-1/2 DVBS2-QPSK-1/3DVBS2-QPSK-1/4 DVBS2-QPSK-2/3 DVBS2-QPSK-3/4 DVBS2-QPSK-2/5 DVBS2-QPSK-3/5 DVBS2-QPSK-4/5 DVBS2-QPSK-5/6 DVBS2-QPSK-8/9 DVBS2-QPSK-9/10 DVBS2-8PSK-2/3 DVBS2-8PSK-3/4 DVBS2-8PSK-3/5 DVBS2-8PSK-5/6 DVBS2-8PSK-8/9 DVBS2-8PSK-9/10 DVBS2-16APSK-2/3 DVBS2-16APSK-3/4 DVBS2-16APSK-4/5 DVBS2-16APSK-5/6 DVBS2-16APSK-8/9 DVBS2-16APSK-9/10 DVBS2-32APSK-3/4 DVBS2-32APSK-4/5 DVBS2-32APSK-5/6 DVBS2-32APSK-8/9 DVBS2-32APSK-9/10 DVBS2X-QPSK-4/15 DVBS2X-QPSK-7/15 DVBS2X-QPSK-8/15 DVBS2X-QPSK-9/20 DVBS2X-QPSK-11/20 DVBS2X-QPSK-11/45 DVBS2X-QPSK-13/45 DVBS2X-QPSK-14/45 DVBS2X-QPSK-32/45 DVBS2X-8PSK-7/15 DVBS2X-8PSK-8/15 DVBS2X-8PSK-13/18 DVBS2X-8PSK-23/36 DVBS2X-8PSK-25/36 DVBS2X-8PSK-26/45 DVBS2X-8PSK-32/45 DVBS2X-8APSK-5/9L DVBS2X-8APSK-26/45L DVBS2X-16APSK-3/5 DVBS2X-16APSK-7/9 DVBS2X-16APSK-7/15 DVBS2X-16APSK-8/15 DVBS2X-16APSK-13/18 DVBS2X-16APSK-23/36 DVBS2X-16APSK-25/36 DVBS2X-16APSK-

name	***type***	***flags***	***range***
			26/45 DVBS2X-16APSK-28/45 DVBS2X-16APSK-32/45 DVBS2X-16APSK-77/90 DVBS2X-16APSK-1/2L DVBS2X-16APSK-2/3L DVBS2X-16APSK-3/5L DVBS2X-16APSK-5/9L DVBS2X-16APSK-8/15L DVBS2X-32APSK-2/3 DVBS2X-32APSK-7/9 DVBS2X-32APSK-11/15 DVBS2X-32APSK-32/45 DVBS2X-32APSK-2/3L DVBS2X-64APSK-4/5 DVBS2X-64APSK-5/6 DVBS2X-64APSK-7/9 DVBS2X-64APSK-11/15 DVBS2X-64APSK-32/45L DVBS2X-128APSK-3/4 DVBS2X-128APSK-7/9 DVBS2X-256APSK-3/4 DVBS2X-256APSK-32/45 DVBS2X-256APSK-2/3L DVBS2X-256APSK-11/15L DVBS2X-256APSK-29/45L DVBS2X-256APSK-31/45L)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
searchRange	INTEGER		IntegerRange (0 .. 5000)
polarization	CHOICE		EnumRange (X Y)
rolloff	CHOICE		EnumRange (35% 25% 20% 15% 10% 5%)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
mis	CHOICE		EnumRange DIS (ON OFF)
isi	INTEGER		IntegerRange DIS (0 .. 255)
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
output.tsOutput	CHOICE		EnumRange (INPUT MPEG2-TRANSCODED DESCRAMBLED)
output.conn1	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
output.conn2	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
output.conn3	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
video.format	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.colorType	TEXT	R/O	StringRange R/O
video.standard	TEXT	R/O	StringRange R/O
video.aspect	TEXT	R/O	StringRange R/O
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.scan	TEXT	R/O	StringRange R/O
video.test	CHOICE		EnumRange (OFF BARS+RED MOVINGBAR BORDER BARS PATHOLOGICAL MULTIBURST LINEUP CONTRAST WHITE)
video.testStandard	CHOICE		EnumRange (AUTO 1080i 30Hz 1080i 29.97Hz 1080i 25Hz 720p 60Hz 720p 59.94Hz 720p 50Hz 480i 29.97Hz 576i 25Hz)
video.input	CHOICE		EnumRange (ASI SAT-VSB ST-INPUT AUTO)
video.fail	CHOICE		EnumRange (FREEZE BLACK BLUE BARS+RED)
video.framesync	CHOICE		EnumRange (OFF ON)
video.convert	CHOICE		EnumRange (AUTO SD HD-1080i HD-720p)

name	*type*	*flags*	*range*
video.rxDelayMode	CHOICE		EnumRange (COMPATIBILITY STANDARD LOW MEGA-LOW USER-DEFINED)
video.625mode	CHOICE		EnumRange (PAL-BDGHI PAL-N PAL-CMB)
video.525mode	CHOICE		EnumRange (NTSC-M PAL-M NTSC-J)
video.trisync	CHOICE		EnumRange (ON ALL ON G/Y NONE)
video.rgbOutput	CHOICE		EnumRange (RGB YPbPr)
video.arConvert	CHOICE		EnumRange (ANAMORPHIC CUTOUT LETTERBOX)
video.arDownConvert	CHOICE		EnumRange (NONE ANAMORPHIC CUTOUT LETTERBOX)
video.embeddedAudio1	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)

name	*type*	*flags*	*range*
video.embeddedAudio2	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio3	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio4	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)

name	*type*	*flags*	*range*
video.embeddedAudio5	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio6	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio7	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)

name	*type*	*flags*	*range*
video.embeddedAudio8	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedOk1	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk2	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk3	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk4	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk5	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk6	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk7	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk8	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.1.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.1.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)

name	*type*	*flags*	*range*
audio.1.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.1.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.1.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.1.source	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.1.clipping	INTEGER	NOPRESET	IntegerRange (0 .. 20)
audio.1.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.2.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.2.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.2.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.2.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.2.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)

name	*type*	*flags*	*range*
audio.2.source	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.2.clipping	INTEGER	NOPRESET	IntegerRange (0 .. 20)
audio.2.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.3.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.3.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.3.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.3.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.3.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)

name	*type*	*flags*	*range*
audio.3.source	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.3.clipping	INTEGER	NOPRESET	IntegerRange (0 .. 20)
audio.3.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.4.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.4.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.4.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.4.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.4.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)

name	*type*	*flags*	*range*
audio.4.source	CHOICE	NOPRESET	EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.4.clipping	INTEGER	NOPRESET	IntegerRange (0 .. 20)
audio.4.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.5.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.5.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.5.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.5.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.5.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.6.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.6.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.6.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.6.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.6.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.6.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.multiProgram	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.scrambled	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.scrambled	CHOICE	R/O	EnumRange R/O (BISS CLEAR)
ca.bissmode	CHOICE		EnumRange (BISS-1 E-FIXED E-TTV E-USER1 E-USER2)
ca.bisskey	TEXT	NOPRESET	StringRange
interface	CHOICE		EnumRange (ASI SAT IP AUTO)
ipin.lastIpRx	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
ipin.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.type	CHOICE	R/O	EnumRange R/O DIS (UDP RTP FEC COLUMNS FEC ROW AND COLUMNS)
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.jitter	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.mdiDelay	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.mdiLossRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
ipin.fecLatency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.bufferLevel	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.fifoOverflow	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.networkUsage	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.networkUsage	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.state.port	CHOICE	R/O	EnumRange R/O DIS (PORT 1 PORT 2 NONE)
ipin.state.stream	CHOICE	R/O	EnumRange R/O DIS (PRIMARY SECONDARY NONE)
ipin.reset	INTEGER		IntegerRange (0 .. 65535)
ipin.bufferDuration	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.1.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange DIS

name	*type*	*flags*	*range*
ipin.1.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.2.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange DIS
ipin.2.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.3.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.3.address	TEXT		StringRange DIS
ipin.3.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.4.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.4.address	TEXT		StringRange DIS
ipin.4.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.interface	CHOICE		EnumRange DIS (PORT 1 PORT 2)
ipin.stream	CHOICE		EnumRange DIS (STREAM 1 STREAM 2)
ipin.failoverTriggerPeriod	INTEGER		IntegerRange DIS (0 .. 999)
ipin.redundancyMode	CHOICE		EnumRange DIS (OFF AUTO TOGGLE)
ipin.failoverMode	CHOICE		EnumRange DIS (PORT ONLY PORT + STREAM)
state.t2.dvbmode	CHOICE	R/O	EnumRange R/O DIS (DVB-T DVB-T2 UNKNOWN)
state.t2.bw	CHOICE	R/O	EnumRange R/O DIS (6MHz 7MHz 8MHz UNKNOWN)
state.t2.preVitBer	TEXT	R/O	StringRange R/O DIS
state.t2.preRsBer	TEXT	R/O	StringRange R/O DIS
state.t2.preLdcpBer	TEXT	R/O	StringRange R/O DIS
state.t2.preBchBer	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
state.t2.mer	TEXT	R/O	StringRange R/O DIS
state.t2.merMargin	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
state.t2.fer	TEXT	R/O	StringRange R/O DIS
state.t2.symbolRate	TEXT	R/O	StringRange R/O DIS
state.t2.temp	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.t2.level	TEXT	R/O	StringRange R/O DIS
t2.fft	CHOICE	R/O	EnumRange R/O DIS (1K 2K 4K 8K 16K 32K 8K-EXT 16K-EXT 32K-EXT UNKNOWN)
t2.guard	CHOICE	R/O	EnumRange R/O DIS (1/32 1/16 1/8 1/4 1/128 19/128 19/256 UNKNOWN)
t2.modulation	CHOICE	R/O	EnumRange R/O DIS (QPSK 0 16-QAM 1 64-QAM 2 256-QAM 3 UNKNOWN 255)
t2.rotation	CHOICE	R/O	EnumRange R/O DIS (NORMAL ROTATED UNKNOWN)
t2.coderate	CHOICE	R/O	EnumRange R/O DIS (1/2 0 3/5 1 2/3 2 3/4 3 4/5 4 5/6 5 UNKNOWN 255)
t2.pilotPattern	CHOICE	R/O	EnumRange R/O DIS (PP1 0 PP2 1 PP3 2 PP4 3 PP5 4 PP6 5 PP7 6 PP8 7 UNKNOWN 255)
t2.papr	CHOICE	R/O	EnumRange R/O DIS (NONE 0 ACE 1 TR 2 TR+ACE 3 UNKNOWN 255)
t2.signalling	CHOICE	R/O	EnumRange R/O DIS (SISO 0 MISO 1 NON-DVB-T2 2 UNKNOWN 255)
t2.fecLength	CHOICE	R/O	EnumRange R/O DIS (SHORT 0 NORMAL 1 UNKNOWN 255)
t2.standard	CHOICE		EnumRange DIS (B/G D/K)

name	*type*	*flags*	*range*
t2.channel	CHOICE		EnumRange DIS (VHF B05 VHF B06 VHF B07 VHF B08 VHF B09 VHF B10 VHF B11 VHF B12 UHF G21 UHF G22 UHF G23 UHF G24 UHF G25 UHF G26 UHF G27 UHF G28 UHF G29 UHF G30 UHF G31 UHF G32 UHF G33 UHF G34 UHF G35 UHF G36 UHF G37 UHF G38 UHF G39 UHF G40 UHF G41 UHF G42 UHF G43 UHF G44 UHF G45 UHF G46 UHF G47 UHF G48 UHF G49 UHF G50 UHF G51 UHF G52 UHF G53 UHF G54 UHF G55 UHF G56 UHF G57 UHF G58 UHF G59 UHF G60 UHF G61 UHF G62 UHF G63)
t2.frequency	FLOAT		DoubleRange DIS (177.000 .. 859.000)
t2.dvbmode	CHOICE		EnumRange DIS (DVB-T DVB-T2 UNKNOWN)
t2.bw	CHOICE		EnumRange DIS (6MHz 7MHz 8MHz DEFAULT UNKNOWN)
t2.longEcho	CHOICE		EnumRange DIS (ON OFF)
t2.forceRetune	CHOICE		EnumRange DIS (ON OFF)
t2.unlockingPeriod	INTEGER		IntegerRange DIS (0 .. 2147483647)
t2.blindTune	CHOICE		EnumRange DIS (CUSTOM AUTO)
t2.searchAll	CHOICE		EnumRange DIS (ON OFF)
t2.ifNotch	CHOICE		EnumRange DIS (OFF ON)
t2.meas	CHOICE		EnumRange DIS (dBuV dBmW)
t2.scan	TEXT	NOPRESET	StringRange
t2.cancelScan	TEXT	NOPRESET	StringRange
t2.retune	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
t2.scan.t	CHOICE		EnumRange DIS (ON OFF)
t2.scan.t2	CHOICE		EnumRange DIS (ON OFF)
t2.scan.start	FLOAT		DoubleRange DIS (177.000 .. 859.000)
t2.scan.stop	FLOAT		DoubleRange DIS (177.000 .. 859.000)
t2.scan.step	FLOAT		DoubleRange DIS (0.100 .. 8.000)
t2.scan.bw6	CHOICE		EnumRange DIS (ON OFF)
t2.scan.bw7	CHOICE		EnumRange DIS (ON OFF)
t2.scan.bw8	CHOICE		EnumRange DIS (ON OFF)
t2.scan.tsCheck	CHOICE		EnumRange DIS (ON OFF)
t2.aScan	BOOLEAN	NOPRESET	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.671 Tandberg-RX8200-TriBand-SNMP

This page describes the device driver and the device window for Tandberg RX8200 DVB-S2 / HDTV IRD, controlled via SNMP.

This is a special version with support for TriBand LNBs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and

BISS keys below.

- --- Contains the parameters concerning the video decoding end video test output.
- --- Contains the parameters concerning the audio outputs and the selection of the audio signals from the MPEG stream. Unlike with other IRD drivers this page also controls the SDI embedded audio assignments.
- --- Contains the parameters concerning the ASI / SDI output ports of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the RX8200.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- 22kHz --- Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
- 22kHzAuto --- If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
- lnbPower --- Selects if the IRD shall apply LNB power to the input.
- lnbVoltage --- Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
- lnbPolSwitch --- If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).

- loFreqHigh --- The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqMid --- The LO frequency for the middle frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- loFreqLow --- The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
- bandEdgeLow --- Below this frequency, the low LO frequency will be selected.
- bandEdgeMid --- Below this frequency (and above bandEdgeLow), the middle LO frequency will be selected.
- spectrumSense --- Controls the spectrum inversion setting of the IRD. AUTO should work in most cases (switches to NORMAL for Ku-band LO Frequencies, INVERTED for C-band frequencies).
- serviceHunt --- If set to ON, the IRD receives the first available program in a stream when tuned to a new signal. With OFF, the IRD insists in receiving the service ID / PIDs which were set before. For SNG hub applications you will usually select ON, for distribution / relaying applications OFF.
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remarks

The RX8200 provides wide functions to monitor the state and control the assignment of SDI embedded audio streams. To present these functions clearly layed out, the embedded audio parameters have been moved to the audio page with this IRD.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
info.lbandInputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.audioChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.swVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqMid	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdgeLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdgeMid	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.spectrumSense	CHOICE	SETUP	EnumRange (AUTO NORMAL INVERTED)
config.serviceHunt	CHOICE	SETUP	EnumRange (OFF ON)
config.marginThreshold	FLOAT	SAVE SETUP	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
config.hasMultiStream	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.ix	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dvbs2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.card	TEXT	R/O	StringRange R/O
internal.ipcard	TEXT	R/O	StringRange R/O
internal.state.ebno	TEXT	R/O	StringRange R/O
internal.state.margin	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
input	CHOICE		EnumRange (1 2 3 4 5 6 7 8)
dvbmode	CHOICE		EnumRange (DVBS DVBS2 AUTO UNKNOWN)
			EnumRange DIS (AUTO DVBS-QPSK-1/2 DVBS-QPSK-2/3 DVBS-QPSK-3/4 DVBS-QPSK-5/6 DVBS-QPSK-6/7 DVBS-QPSK-7/8 DVBS-8PSK-2/3 DVBS-8PSK-5/6 DVBS-8PSK-8/9 DVBS-8PSK-16QAM-3/4 DVBS-8PSK-16QAM-7/8 DVBS2-QPSK-1/2 DVBS2-QPSK-1/3DVBS2-QPSK-1/4 DVBS2-QPSK-2/3 DVBS2-QPSK-3/4 DVBS2-QPSK-2/5 DVBS2-QPSK-3/5 DVBS2-QPSK-4/5 DVBS2-QPSK-5/6 DVBS2-QPSK-8/9 DVBS2-QPSK-9/10 DVBS2-8PSK-2/3 DVBS2-8PSK-3/4 DVBS2-8PSK-3/5 DVBS2-8PSK-5/6 DVBS2-8PSK-8/9 DVBS2-8PSK-9/10 DVBS2-16APSK-2/3 DVBS2-16APSK-3/4 DVBS2-16APSK-4/5 DVBS2-16APSK-5/6 DVBS2-16APSK-8/9 DVBS2-16APSK-9/10 DVBS2-

name	*type*	*flags*	*range*
modCod	CHOICE		32APSK-3/4 DVBS2-32APSK-4/5 DVBS2-32APSK-5/6 DVBS2-32APSK-8/9 DVBS2-32APSK-9/10 DVBS2X-QPSK-4/15 DVBS2X-QPSK-7/15 DVBS2X-QPSK-8/15 DVBS2X-QPSK-9/20 DVBS2X-QPSK-11/20 DVBS2X-QPSK-11/45 DVBS2X-QPSK-13/45 DVBS2X-QPSK-14/45 DVBS2X-QPSK-32/45 DVBS2X-8PSK-7/15 DVBS2X-8PSK-8/15 DVBS2X-8PSK-13/18 DVBS2X-8PSK-23/36 DVBS2X-8PSK-25/36 DVBS2X-8PSK-26/45 DVBS2X-8PSK-32/45 DVBS2X-8APSK-5/9L DVBS2X-8APSK-26/45L DVBS2X-16APSK-3/5 DVBS2X-16APSK-7/9 DVBS2X-16APSK-7/15 DVBS2X-16APSK-8/15 DVBS2X-16APSK-13/18 DVBS2X-16APSK-23/36 DVBS2X-16APSK-25/36 DVBS2X-16APSK-26/45 DVBS2X-16APSK-28/45 DVBS2X-16APSK-32/45 DVBS2X-16APSK-77/90 DVBS2X-16APSK-1/2L DVBS2X-16APSK-2/3L DVBS2X-16APSK-3/5L DVBS2X-16APSK-5/9L DVBS2X-16APSK-8/15L DVBS2X-32APSK-2/3 DVBS2X-32APSK-7/9 DVBS2X-32APSK-11/15 DVBS2X-32APSK-32/45 DVBS2X-32APSK-2/3L DVBS2X-64APSK-4/5 DVBS2X-64APSK-5/6 DVBS2X-64APSK-7/9 DVBS2X-64APSK-11/15 DVBS2X-64APSK-32/45L DVBS2X-128APSK-3/4 DVBS2X-128APSK-7/9 DVBS2X-256APSK-3/4 DVBS2X-256APSK-32/45 DVBS2X-256APSK-2/3L DVBS2X-256APSK-11/15L DVBS2X-256APSK-29/45L DVBS2X-256APSK-31/45L)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
modulation	TEXT	R/O	StringRange R/O
fec	TEXT	R/O	StringRange R/O
searchRange	INTEGER		IntegerRange (0 .. 5000)
polarization	CHOICE		EnumRange (X Y)
rolloff	CHOICE		EnumRange (35% 25% 20% 15% 10% 5%)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O
mis	CHOICE		EnumRange DIS (ON OFF)
isi	INTEGER		IntegerRange DIS (0 .. 255)
programCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
output.tsOutput	CHOICE		EnumRange (INPUT MPEG2-TRANSCODED DESCRAMBLED)
output.conn1	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
output.conn2	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
output.conn3	CHOICE		EnumRange (AUTO ASI SD-SDI HD-SDI SDI-3G)
video.format	TEXT	R/O	StringRange R/O
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 0)
video.bitrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
video.colorType	TEXT	R/O	StringRange R/O
video.standard	TEXT	R/O	StringRange R/O
video.aspect	TEXT	R/O	StringRange R/O
video.framerate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
video.scan	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
video.test	CHOICE		EnumRange (OFF BARS+RED MOVINGBAR BORDER BARS PATHOLOGICAL MULTIBURST LINEUP CONTRAST WHITE)
video.testStandard	CHOICE		EnumRange (AUTO 1080i 30Hz 1080i 29.97Hz 1080i 25Hz 720p 60Hz 720p 59.94Hz 720p 50Hz 480i 29.97Hz 576i 25Hz)
video.input	CHOICE		EnumRange (ASI SAT-VSB ST-INPUT AUTO)
video.fail	CHOICE		EnumRange (FREEZE BLACK BLUE BARS+RED)
video.framesync	CHOICE		EnumRange (OFF ON)
video.convert	CHOICE		EnumRange (AUTO SD HD-1080i HD-720p)
video.rxDelayMode	CHOICE		EnumRange (COMPATIBILITY STANDARD LOW MEGA-LOW USER-DEFINED)
video.625mode	CHOICE		EnumRange (PAL-BDGHI PAL-N PAL-CMB)
video.525mode	CHOICE		EnumRange (NTSC-M PAL-M NTSC-J)
video.trisync	CHOICE		EnumRange (ON ALL ON G/Y NONE)
video.rgbOutput	CHOICE		EnumRange (RGB YPbPr)
video.arConvert	CHOICE		EnumRange (ANAMORPHIC CUTOUT LETTERBOX)
video.arDownConvert	CHOICE		EnumRange (NONE ANAMORPHIC CUTOUT LETTERBOX)

name	*type*	*flags*	*range*
video.embeddedAudio1	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio2	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio3	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)

name	*type*	*flags*	*range*
video.embeddedAudio4	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio5	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio6	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)

name	*type*	*flags*	*range*
video.embeddedAudio7	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedAudio8	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-EMBEDDING 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
video.embeddedOk1	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk2	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk3	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk4	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk5	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk6	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk7	BOOLEAN	R/O	BooleanRange R/O (true false)
video.embeddedOk8	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)

name	*type*	*flags*	*range*
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.1.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.coding	TEXT	R/O	StringRange R/O
audio.1.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.1.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.1.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.1.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.1.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.1.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.1.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.1.source	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.1.clipping	INTEGER		IntegerRange (0 .. 20)
audio.1.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.2.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.coding	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audio.2.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.2.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.2.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.2.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.2.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.2.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.2.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.2.source	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.2.clipping	INTEGER		IntegerRange (0 .. 20)
audio.2.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.3.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.3.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.coding	TEXT	R/O	StringRange R/O
audio.3.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.3.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.3.gain	FLOAT		DoubleRange (-6.0 .. 6.0)

name	*type*	*flags*	*range*
audio.3.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.3.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.3.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.3.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.3.source	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.3.clipping	INTEGER		IntegerRange (0 .. 20)
audio.3.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.4.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.4.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.coding	TEXT	R/O	StringRange R/O
audio.4.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.4.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.4.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.4.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.4.delay	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
audio.4.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.4.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.4.source	CHOICE		EnumRange (1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-OFF NO-OUTPUT 1-STEREO 2-STEREO 3-STEREO 4-STEREO 5-STEREO 6-STEREO 1-DOWNMIX 2-DOWNMIX 3-DOWNMIX 4-DOWNMIX 5-DOWNMIX 6-DOWNMIX 1-LR 2-LR 3-LR 4-LR 5-LR 6-LR 1-C-LFE 2-C-LFE 3-C-LFE 4-C-LFE 5-C-LFE 6-C-LFE 1-LS-RS 2-LS-RS 3-LS-RS 4-LS-RS 5-LS-RS 6-LS-RS 1-COMP-AUD 2-COMP-AUD 3-COMP-AUD 4-COMP-AUD 5-COMP-AUD 6-COMP-AUD)
audio.4.clipping	INTEGER		IntegerRange (0 .. 20)
audio.4.outOk	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)
audio.5.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.5.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.coding	TEXT	R/O	StringRange R/O
audio.5.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.5.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.5.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.5.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.5.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.5.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.5.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
audio.6.pid	CHOICE	NOPRESET	EnumRange (NO SELECTION)

name	*type*	*flags*	*range*
audio.6.running	BOOLEAN	R/O	BooleanRange R/O (true false)
audio.6.bitrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.coding	TEXT	R/O	StringRange R/O
audio.6.sampling	INTEGER	R/O	IntegerRange R/O (0 .. 0)
audio.6.downmix	CHOICE		EnumRange (SURROUND STEREO)
audio.6.gain	FLOAT		DoubleRange (-6.0 .. 6.0)
audio.6.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.6.delay	FLOAT		DoubleRange (0.0 .. 0.0)
audio.6.dynamic	CHOICE		EnumRange (OFF LINE-MODE RF-MODE)
audio.6.lineAdjust	CHOICE		EnumRange (OFF LOW HIGH)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ts	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.asi	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.multiProgram	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.scrambled	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.scrambled	CHOICE	R/O	EnumRange R/O (BISS CLEAR)
ca.bissmode	CHOICE		EnumRange (BISS-1 E-FIXED E-TTV E-USER1 E-USER2)
ca.bisskey	TEXT		StringRange

name	*type*	*flags*	*range*
interface	CHOICE		EnumRange (ASI SAT IP AUTO)
ipin.lastIpRx	TEXT	R/O	StringRange R/O DIS
ipin.fecColumns	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.fecRows	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsFrame	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.type	CHOICE	R/O	EnumRange R/O DIS (UDP RTP FEC COLUMNS FEC ROW AND COLUMNS)
ipin.packetsRx	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsCorr	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.packetsLost	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.jitter	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.mdiDelay	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.mdiLossRate	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
ipin.fecLatency	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.bufferLevel	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.fifoOverflow	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.1.networkUsage	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.2.networkUsage	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
ipin.state.port	CHOICE	R/O	EnumRange R/O DIS (PORT 1 PORT 2 NONE)
ipin.state.stream	CHOICE	R/O	EnumRange R/O DIS (PRIMARY SECONDARY NONE)
ipin.reset	INTEGER		IntegerRange (0 .. 65535)
ipin.bufferDuration	INTEGER		IntegerRange DIS (0 .. 65535)
ipin.fecEnable	CHOICE		EnumRange DIS (OFF ON)
ipin.1.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)

name	*type*	*flags*	*range*
ipin.1.address	TEXT		StringRange DIS
ipin.1.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.2.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange DIS
ipin.2.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.3.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.3.address	TEXT		StringRange DIS
ipin.3.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.4.mode	CHOICE		EnumRange DIS (MULTICAST UNICAST)
ipin.4.address	TEXT		StringRange DIS
ipin.4.port	INTEGER		IntegerRange DIS (0 .. 0)
ipin.interface	CHOICE		EnumRange DIS (PORT 1 PORT 2)
ipin.stream	CHOICE		EnumRange DIS (STREAM 1 STREAM 2)
ipin.failoverTriggerPeriod	INTEGER		IntegerRange DIS (0 .. 999)
ipin.redundancyMode	CHOICE		EnumRange DIS (OFF AUTO TOGGLE)
ipin.failoverMode	CHOICE		EnumRange DIS (PORT ONLY PORT + STREAM)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Transport stream)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Margin)

9.672 Tandberg-SM62xx

This page describes the device driver and the device window for the Tandberg SM62XX DVB modulator family.

It was tested with the SM 6620 SW 04.17.

It supports also SM6610, SM6615 and 6625. Therefore the model can be adapted using the config.model setup parameter

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- RF / modulator parameters
- --- Device fault list messages
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the encoder.

parameter	***description***
address	Enter the IP address of the encoder here, followed by the community string 'private'. Example: **10.0.0.1 private**
frontPanel	Switches front panel between full control and monitor only
freqMode	Switches between usage of IF and RF frequency
model	Allows to select the appropriate model
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.swrelease	TEXT	R/O	StringRange R/O
info.serialno	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.frontPanel	CHOICE	SETUP	EnumRange (FULL MONITOR)
config.freqMode	CHOICE	SAVE SETUP	EnumRange (IF RF)
config.model	CHOICE	SAVE SETUP	EnumRange (SM6610 SM6615 SM6620 SM6625)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input.source	CHOICE		EnumRange (ASI-1 ASI-2 SPI NULL PRBS ASI-3 ASI-4)
input.packetsize	CHOICE		EnumRange (188 204 UNFRAMED)
input.datarate	FLOAT		DoubleRange (0.711130 .. 213.000000)
mod.power	FLOAT		DoubleRange (-20.0 .. 5.0)
mod.symbolrate	FLOAT		DoubleRange (2.000000 .. 66.000000)
mod.standard	CHOICE		EnumRange (DVB-S DVB-S2)
mod.type	CHOICE		EnumRange (BPSK QPSK 8PSK 16QAM 16APSK 32APSK)
mod.fec	CHOICE		EnumRange (1/2 2/3 3/4 4/5 5/6 7/8 8/9 9/10 1/4 1/3 2/5 3/5)

name	*type*	*flags*	*range*
mod.rollOff	CHOICE		EnumRange (20% 25% 30% 35%)
mod.rfon	CHOICE		EnumRange (OFF ON)
mod.modulation	CHOICE		EnumRange (OFF ON)
mod.spectrum	CHOICE		EnumRange (NORM INVERT)
mod.rateAdapter	CHOICE		EnumRange (OFF ON)
mod.frequency	FLOAT		DoubleRange (50.000 .. 180.000)
mod.pilot	CHOICE		EnumRange (OFF ON)
mod.frameSize	CHOICE		EnumRange (NORMAL SHORT)
mod.mapMode	CHOICE		EnumRange (PEAK PWR MEAN PWR)
mod.ifTilt	FLOAT		DoubleRange (-0.040 .. 0.040)
mod.goldSeqNum	INTEGER		IntegerRange (0 .. 262141)
mod.upconverter	CHOICE		EnumRange (OFF ON)
mod.reference	CHOICE		EnumRange (OFF ON)
faultList	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Host Controller Alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Host Controller Warn)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input Card Alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input Card Warning)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Baseband Alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Baseband Warning)
faults.07	ALARM	R/O	AlarmFlagRange R/O (IF Card Alarm)
faults.08	ALARM	R/O	AlarmFlagRange R/O (IF Card Warning)

9.673 Tandberg-TT1260-SNMP

This page describes the device driver and the device window for Tandberg TT1260 IRD,

controlled via SNMP. The actual version of the driver supports the TT1260 with 2 RF inputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video and audio output.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the TT1260.

parameter	***description***
address	Enter the IP address of the IRD here, in 'dotted quad' notation. The IRD requires the community string 'private' to be appended to the address. Example **10.0.0.1 private**
lbandInputs	This parameter should be set to the number of L-band input the IRD is equipped with (1/2/4).
22kHz	Makes the IRD apply a 22 kHz tone to the LNB power when set to ON.
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Selects if the IRD shall apply LNB power to the input.
lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
lnbPolSwitch	If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).

parameter	*description*
lnbBoost	Raises the LNB supply voltage by 1V if set to ON (long cable compensation).
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputcard	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (4 2 1)
config.22kHz	CHOICE	SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbBoost	CHOICE	SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
internal.ix	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dvbs2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input	CHOICE		EnumRange (1 2 3 4 5 6)
dvbmode	CHOICE		EnumRange DIS (DVBS DVBS2)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
modulation	TEXT	R/O	StringRange R/O
fec	CHOICE		EnumRange (NONE 1/2 2/3 3/4 5/6 6/7 7/8 8/9 AUTO)
searchRange	INTEGER		IntegerRange (0 .. 5000)
polarization	CHOICE		EnumRange (X Y)

name	*type*	*flags*	*range*
rolloff	CHOICE		EnumRange DIS (35% 25% 20%)
programNo	INTEGER		IntegerRange (0 .. 65535)
programList	CHOICE		EnumRange (NO SERVICES)
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
service.videoPID	INTEGER		IntegerRange (0 .. 8191)
service.pcrPID	INTEGER		IntegerRange (0 .. 8191)
video.test	CHOICE		EnumRange (OFF PAL-1 PAL-2 PAL-3 PAL-4 PAL-5 PAL-6 PAL-7 PAL-8 PAL-9 PAL-10 NTSC-1 NTSC-2 NTSC-3 NTSC-4 NTSC-5 NTSC-6 NTSC-7 NTSC-8 NTSC-9 NTSC-10)
video.fail	CHOICE		EnumRange (FREEZE BLACK NO SYNC RED+TEXT)
video.framesync	CHOICE		EnumRange (OFF ON)
video.mode	CHOICE		EnumRange (PAL-625 NTSC-525)
video.625mode	CHOICE		EnumRange (PAL-BGI PAL-N PAL-NCMB)
video.525mode	CHOICE		EnumRange (NTSC-M PAL-M PAL-525)
video.aspect	CHOICE		EnumRange (4/3 16/9)
video.colorCoding	CHOICE		EnumRange (UNKNOWN 4:2:0 4:2:2)
audio.1.program	INTEGER		IntegerRange (0 .. 8191)
audio.1.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.1.output	CHOICE		EnumRange (AES AC3)
audio.1.level	INTEGER		IntegerRange (12 .. 24)
audio.1.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.1.delay	INTEGER		IntegerRange (-49 .. 49)
audio.1.running	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
audio.2.program	INTEGER		IntegerRange (0 .. 8191)
audio.2.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.2.output	CHOICE		EnumRange (AES AC3)
audio.2.level	INTEGER		IntegerRange (12 .. 24)
audio.2.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.2.delay	INTEGER		IntegerRange (-49 .. 49)
audio.2.running	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.margin	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.level	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
state.plen	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ca.bissmode	CHOICE		EnumRange (BISS-1 E-FIXED E-TTV E-USER1 E-USER2)
ca.rasmode	CHOICE		EnumRange (FIXED DSNG)
ca.bisskey	TEXT		StringRange
ca.raskey	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)

9.674 Tandberg-TT1260

This page describes the device driver and the device window for Tandberg TT1260 IRD. The actual version of the driver supports the TT1260 with a QPSK demodulator an 1 to 4 inputs. The

sat-nms software controls a TT1260 via it's serial (RS232) interface using the Tandberg TT-protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the conditional access parameters. See also the remark about RAS and BISS keys below.
- --- Contains the parameters concerning the video and audio output.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the TT1260.

parameter	***description***
address	The TT1260 communication protocol does not use a device address. leave this field empty.
lbandInputs	Select the number of L-band inputs the receiver provides (depending on the model 1, 2 or 4 inputs)
22kHz	Makes the IRD apply a 22 kHz tone to the LNB power for frequency settings above the 'bandEdge' frequency if set to 'AUTO'. With the setting 'OFF' no 22 kHz tone is generated. Changing this value will change the setting for the actually selected input. Whenever the input is switched, the setting defined here is applied to the active input. If you change the frequency, the 22 kHz is applied to the currently active input depending on the new frequency value. You need to switch on the LNB power if you want to use the 22 kHz tone.
lnbPower	Selects if the IRD shall apply LNB power to the actually selected input. Changing this value will change the setting for the actually selected input. Whenever the input is switched, the setting defined here is applied to the active input.
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

parameter	*description*
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies (and the 22 kHz tone if enabled).
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the IRD. The proper setting is *38400,N,8,1* .

Receiving encrypted programs

Depending on the TT1260 model you own, the IRD is capable to decrypt RAS and/or BISS encoded streams. To make the IRD decrypt these signals properly, please consider the following:

- For some BISS-E modes the user-1 and user-2 keys must be set at the IRD. This can be done at the front panel only, there is no remote control for these keys foreseen.
- The BISS session key is entered at the sat-nms user interface. Depending on the BISS mode, this key is to be entered as a 48 bit (12 digit) or 64 bit (16 digit) hexadecimal number. The software accepts both, upper case and lower case letters for the digits A .. F. The key cannot be read back, the software shows the text '-- HIDDEN --' in the entry field after the IRD has accepted the key.
- RAS (DSNG) keys are to be entered a 5 digit decimal numbers. The software displays the key which has been set recently at the IRD.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.hwrevision	TEXT	R/O	StringRange R/O
config.lbandInputs	CHOICE	SAVE SETUP	EnumRange (1 2 4)
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF AUTO)
config.lnbPower	CHOICE	SETUP	EnumRange (OFF 14V-Vertical 18V-Horizontal)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.band	TEXT	R/O	StringRange R/O
internal.biss	TEXT	R/O	StringRange R/O
input	CHOICE		EnumRange (1)
frequency	FLOAT		DoubleRange (0.0 .. 0.0)
dataRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (1.000 .. 46.000)
modulation	CHOICE		EnumRange (QPSK)
fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8 AUTO)
programNo	INTEGER		IntegerRange (0 .. 65535)
searchRange	INTEGER		IntegerRange (0 .. 5000)
programList	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.test	CHOICE		EnumRange (OFF PULS&BAR MLTIBRST PULS&STP MAGENTSTP FLATRED LUMRAMP YUVRAMP C-BARS MLTIPAT BLACK NTSC-1 NTSC-2 NTSC-3 NTSC-4 NTSC-5 NTSC-6 NTSC-7 NTSC-8 NTSC-9 NTSC-10)
video.fail	CHOICE		EnumRange (FREEZE BLACK NO SYNC)
video.framesync	CHOICE		EnumRange (OFF ON)
video.mode	CHOICE		EnumRange (PAL-625 NTSC-525)
video.625mode	CHOICE		EnumRange (PAL-BGI PAL-N PAL-NCMB)
video.525mode	CHOICE		EnumRange (NTSC-M NTSC-MNP PAL-M)
audio.1.program	INTEGER		IntegerRange (0 .. 8191)
audio.1.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.1.output	CHOICE		EnumRange (AES AC3)
audio.1.level	INTEGER		IntegerRange (12 .. 24)
audio.1.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.1.delay	INTEGER		IntegerRange (-49 .. 49)
audio.2.program	INTEGER		IntegerRange (0 .. 8191)
audio.2.routing	CHOICE		EnumRange (STEREO MONO LEFT RIGHT)
audio.2.output	CHOICE		EnumRange (AES AC3)
audio.2.level	INTEGER		IntegerRange (12 .. 24)
audio.2.ac3mix	CHOICE		EnumRange (SURROUND STEREO)
audio.2.delay	INTEGER		IntegerRange (-49 .. 49)
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ber	TEXT	R/O	StringRange R/O
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.aspect	TEXT	R/O	StringRange R/O
state.state	TEXT	R/O	StringRange R/O
ca.bissmode	CHOICE		EnumRange (MODE-1 E-FIXED E-TTV E-USER1 E-USER2)
ca.rasmode	CHOICE		EnumRange (FIXED DSNG)
ca.bisskey	TEXT		StringRange
ca.raskey	INTEGER		IntegerRange (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)

9.675 Technotrans-HC-Vertex

This page describes the device driver and the device window for a technotrans air condition controller type HC-Vertex. It allows reading temperature, airflow, valves as well as adjustment of their setpoints.

The driver uses the Modbus TCP protocol. The default TCP port used is 502.

It was tested with the following SW release:

- Software 90000488481 Version 0.92 Identifier Vertex
- Firmware 1.23.45 Version 0.80 Identifier CP 1382

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- System Overview and measurement values.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must always be set to 1 if the device is controlled Modbus over a serial line. With Modbus/TCP the device address doesn't matter.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
highcontrol.1	CHOICE	R/O	EnumRange R/O (SLAVE MASTER)
highcontrol.2	CHOICE	R/O	EnumRange R/O (SLAVE MASTER)
highcontrol	CHOICE	R/O	EnumRange R/O (SLAVE MASTER)
scu.1.tempHub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.1.tempHubF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.1.tempHpa	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.1.tempHpaF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
scu.1.tempTank	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.1.tempTankF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.1.highPressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempHub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempHubF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempHpa	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempHpaF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempTank	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.tempTankF	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
scu.2.highPressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.1.flow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.2.flow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.3.flow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.1.slaveFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.2.slaveFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.3.slaveFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.1.diffFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.2.diffFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hpa.3.diffFlow	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.flow	FLOAT		DoubleRange (2.0 .. 6.0)
flags.flowHpa1	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.flowHpa2	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.flowHpa3	BOOLEAN	R/O	BooleanRange R/O (true false)
setpoint.tempHub	FLOAT		DoubleRange (20.0 .. 34.0)
setpoint.tempHpa	FLOAT		DoubleRange (20.0 .. 42.0)
setpoint.tempTank	FLOAT	R/O	DoubleRange R/O (13.0 .. 35.1)
setpoint.maintenance	FLOAT		DoubleRange (20.0 .. 35.0)

name	*type*	*flags*	*range*
scu.1.valve	INTEGER	R/O	IntegerRange R/O (0 .. 100)
scu.2.valve	INTEGER	R/O	IntegerRange R/O (0 .. 100)
hpa.1.valve	INTEGER	R/O	IntegerRange R/O (0 .. 100)
hpa.2.valve	INTEGER	R/O	IntegerRange R/O (0 .. 100)
hpa.3.valve	INTEGER	R/O	IntegerRange R/O (0 .. 100)
scu.1.state	CHOICE	R/O	EnumRange R/O (COOLING STANDBY)
scu.2.state	CHOICE	R/O	EnumRange R/O (COOLING STANDBY)
switchtime.hrs	INTEGER		IntegerRange (0 .. 24)
switchtime.min	INTEGER		IntegerRange (0 .. 59)
maintenance	CHOICE		EnumRange (ON OFF)
state.maintenance	CHOICE	R/O	EnumRange R/O (NOT ACTIVE ACTIVATED ACTIVE DEACTIVATED)
switchOver	CHOICE		EnumRange (SWITCHED NOT SWITCHED)
reInit	CHOICE	NOPRESET	EnumRange (0 1)
switch	CHOICE	NOPRESET	EnumRange (0 1)
switch.daily	CHOICE		EnumRange (ON OFF)
operatingMode	CHOICE		EnumRange (COOLING HEATING)
info.minTempTankSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.maxTempTankSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.minTempHubSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.maxTempHubSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.minTempHpaSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.maxTempHpaSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.minFlowSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.maxFlowSet	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
maxTempAlarmHub	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
maxTempAlarmHpa	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
maxTempAlarmTank	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
minTempAlarmTank	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
setpoint.tempTankCool	FLOAT		DoubleRange (13.0 .. 35.1)
setpoint.tempTankHeat	FLOAT		DoubleRange (13.0 .. 35.1)
setpoint.minFlowAlarm	FLOAT		DoubleRange (1.0 .. 4.0)
setpoint.maxFlowAlarm	FLOAT		DoubleRange (4.0 .. 8.0)
setpoint.minTempAlarm	FLOAT		DoubleRange (1.0 .. 10.0)
setpoint.maxTempAlarm	FLOAT		DoubleRange (1.0 .. 10.0)
setpoint.highPressure	FLOAT		DoubleRange (15.0 .. 25.0)
setpoint.deltaJointIn	FLOAT		DoubleRange (2.0 .. 20.0)
scu.1.waterPressure	FLOAT		DoubleRange (0.0 .. 0.0)
scu.2.waterPressure	FLOAT		DoubleRange (0.0 .. 0.0)
setpoint.minWaterWarn	FLOAT		DoubleRange (0.0 .. 2.5)
switchOverWaterPress	FLOAT		DoubleRange (0.0 .. 1.5)
config.controller	CHOICE	SAVE SETUP	EnumRange (1 2)
config.hasWaterPressSensor	CHOICE	SAVE SETUP	EnumRange (NO YES)
internal.1	CHOICE	R/O	EnumRange R/O (SLAVE MASTER)
internal.2	CHOICE	R/O	EnumRange R/O (SLAVE MASTER)
internal.sw1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sw2	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tower A/C 1)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Tower A/C 2)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HC System)
faults.06	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Summary)
faults.07	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Summary)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Valve Cluster)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Maintenance)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Hub Door Open)
faults.11	ALARM	R/O	AlarmFlagRange R/O (SCU 1 High Pressure)
faults.27	ALARM	R/O	AlarmFlagRange R/O (SCU 2 High Pressure)
faults.12	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Low Pressure)
faults.28	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Low Pressure)
faults.13	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Compressor)
faults.29	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Compressor)
faults.14	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Pumps)
faults.30	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Pumps)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Internal Fan)
faults.31	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Internal Fan)
faults.16	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Heater)
faults.32	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Heater)
faults.17	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Low Water)
faults.33	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Low Water)
faults.18	ALARM	R/O	AlarmFlagRange R/O (SCU 1 No Flow)
faults.34	ALARM	R/O	AlarmFlagRange R/O (SCU 2 No Flow)
faults.19	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Pump off)
faults.35	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Pump off)
faults.20	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Temp Hub)
faults.36	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Temp Hub)
faults.21	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Temp HPA)
faults.37	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Temp HPA)
faults.22	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Temp Tank)
faults.38	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Temp Tank)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (SCU 1 High Press Warn)
faults.39	ALARM	R/O	AlarmFlagRange R/O (SCU 2 High Press Warn)
faults.64	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Compr.-Heater)
faults.65	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Compr.-Heater)
faults.66	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Min/Max Op. Temp.)
faults.67	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Min/Max Op. Temp.)
faults.24	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Comms HC)
faults.40	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Comms HC)
faults.25	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Ctrl High Press)
faults.41	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Ctrl High Press)
faults.26	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Ext Temp Sensor)
faults.42	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Ext Temp Sensor)
faults.43	ALARM	R/O	AlarmFlagRange R/O (HC 1 COM Error)
faults.59	ALARM	R/O	AlarmFlagRange R/O (HC 2 COM Error)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Fan Fuse Hub 1)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Fan Fuse Hub 2)

name	*type*	*flags*	*range*
faults.45	ALARM	R/O	AlarmFlagRange R/O (HC 1 Flowmeter HPA 1)
faults.61	ALARM	R/O	AlarmFlagRange R/O (HC 2 Flowmeter HPA 1)
faults.46	ALARM	R/O	AlarmFlagRange R/O (HC 1 Flowmeter HPA 2)
faults.62	ALARM	R/O	AlarmFlagRange R/O (HC 2 Flowmeter HPA 2)
faults.47	ALARM	R/O	AlarmFlagRange R/O (HC 1 Flowmeter HPA 3)
faults.63	ALARM	R/O	AlarmFlagRange R/O (HC 2 Flowmeter HPA 3)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Max Flow HPA 1)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Max Flow HPA 2)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Max Flow HPA 3)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Min Flow HPA 1)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Min Flow HPA 2)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Min Flow HPA 3)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Flow Not Plausibe HPA 1)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Flow Not Plausibe HPA 2)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Flow Not Plausibe HPA 3)
faults.57	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Flowswitch)

name	*type*	*flags*	*range*
faults.58	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Flowswitch)
faults.68	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Water Press Sens)
faults.69	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Water Press Sens)
faults.70	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Low Water Press)
faults.71	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Low Water Press)
faults.72	ALARM	R/O	AlarmFlagRange R/O (SCU 1 Water Press Switch)
faults.73	ALARM	R/O	AlarmFlagRange R/O (SCU 2 Water Press Switch)

9.676 Teledyne-QFlex-400

This page describes the device driver and the device window for the Teledyne-QFlex-400 device. It supports modem functionalities with RF rx and tx interface as well as four ethernet interfaces for traffic. This driver only controls the RF functionalities of the modem. Source and destination IP addresses have to be set manually in the web based interface.

Known issues

When setting or changing the modulation or reloading a stored configuration, the device itself takes up to several seconds to process these changes and is unable. Communication faults that last a few seconds are expected when performing some changes due to the nature of the QFlex-400 device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Lets the user set the modem type as well as TX/RX-frequencies and power.
- --- Contains the parameters for the TX modulator such as data-/symbolrate, modulation, fec etc.
- --- Contains similar parameters but for demodulation.
- --- Displays the current bit- and datarate, dropped packets and error packets for tx and

rx.

- --- Displays RF receive status e.g. eb/n0 or power level.
- --- Enables the saving and recalling internally stored device configurations.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- pupPassword --- The password for using PUP Commands is to be set here as a string. The internal PUP Command "console" is used by the driver to save configurations in the device internally, delete them or acces them to be applicated.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.modeltype	TEXT	R/O	StringRange R/O
info.serialnumber	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.fwrevision	TEXT	R/O	StringRange R/O
info.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 55.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.pupPassword	TEXT	SAVE SETUP	StringRange
tx.modemType	CHOICE	NOPRESET	EnumRange (OFF DVBS2 DVBS2X CLOSED NETWORK CLOSED NETWORK PLUS ESC)
tx.on	CHOICE	NOPRESET	EnumRange (OFF ON ON(MUTE IF PWR BREAK) RTS ENABLED RX ENABLED ON(MUTE IF LOSS OF M&C))
tx.power	FLOAT		DoubleRange (-45.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.band	CHOICE		EnumRange (L-BAND IF)
tx.mod.rateControl	CHOICE		EnumRange (DATARATE SYMBOLRATE)
tx.mod.dataRate	FLOAT		DoubleRange (0.002400 .. 60.000000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.002400 .. 50.000000)

name	*type*	*flags*	*range*
tx.mod.pilots	CHOICE		EnumRange (ON OFF)
tx.mod.fecFramesize	CHOICE		EnumRange (NORMAL SHORT)
tx.mod.fecType	CHOICE		EnumRange (OFF VITERBI TPC SMARTLINK)
tx.mod.fec	CHOICE		EnumRange (0.312(5/16) 0.477(21/44) 0.493(1/2) 1/2 2/3 0.666(2/3) 3/4 0.75(3/4) 0.789 7/8 0.875(7/8) 0.875(7/8) P r15240-16404 1/4 1/3 2/5 3/5 4/5 5/6 8/9 9/10 11/45 4/15 13/45 14/45 9/20 7/15 8/15 11/20 26/45 28/45 23/36 25/36 32/45 13/18 11/15 7/9 77/90 23/26 1/2-L 8/15-L 5/9-L 26/45-L 3/5-L 2/3-L 32/45-L 4/9 5/9 12/15 29/45-L 31/45-L 11/15-L)
tx.mod.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16QAM 8QAM 16APSK 32APSK 64QAM 64APSK 8APSK 128APSK 256APSK)
tx.mod.testPattern	CHOICE		EnumRange (OFF MODULAR CW MODULAR ALTERNATE)
tx.mod.protection	CHOICE		EnumRange (ON OFF)
tx.mod.clockSource	CHOICE		EnumRange (EXTERNAL INTERNAL RX REFERENCE STATION)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORMAL INVERTED)
tx.mod.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
rx.modemType	CHOICE	NOPRESET	EnumRange (OFF DVBS2 DVBS2X CLOSED NETWORK CLOSED NETWORK PLUS ESC)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.tracktx	CHOICE		EnumRange (ON OFF)
rx.band	CHOICE		EnumRange (L-BAND IF)

name	*type*	*flags*	*range*
rx.dem.rateControl	CHOICE		EnumRange (DATARATE SYMBOLRATE)
rx.dem.dataRate	FLOAT		DoubleRange (0.004800 .. 52.000000)
rx.dem.symbolRate	FLOAT		DoubleRange (0.009600 .. 10.000000)
rx.dem.pilots	CHOICE		EnumRange (ON OFF)
rx.dem.fecFramesize	CHOICE		EnumRange (NORMAL SHORT)
rx.dem.fecType	CHOICE		EnumRange (OFF VITERBI TPC SMARTLINK)
rx.dem.fec	CHOICE		EnumRange (0.312(5/16) 0.477(21/44) 0.493(1/2) 1/2 2/3 0.666(2/3) 3/4 0.75(3/4) 0.789 7/8 0.875(7/8) 0.875(7/8) P r15240-16404 1/4 1/3 2/5 3/5 4/5 5/6 8/9 9/10 11/45 4/15 13/45 14/45 9/20 7/15 8/15 11/20 26/45 28/45 23/36 25/36 32/45 13/18 11/15 7/9 77/90 23/26 1/2-L 8/15-L 5/9-L 26/45-L 3/5-L 2/3-L 32/45-L 4/9 5/9 12/15 29/45-L 31/45-L 11/15-L)
rx.dem.type	CHOICE		EnumRange (BPSK QPSK OQPSK 8PSK 16QAM 8QAM 16APSK 32APSK 64QAM 64APSK 8APSK 128APSK 256APSK)
rx.dem.clockSource	CHOICE		EnumRange (SATELLITE)
rx.dem.spectrumInvert	CHOICE		EnumRange (NORMAL INVERTED)
rx.dem.sweepMode	CHOICE		EnumRange (NORMAL CUSTOM)
rx.dem.sweepRange	FLOAT		DoubleRange (1 .. 250)
rx.dem.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
loopBack	CHOICE		EnumRange (OFF INTERFACE INTERFACE(LOCAL+REMOTE))
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
rx.meter.power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.freqoffs	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.basebandframes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.compositepower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.meter.basebanderrors	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.meter.bufferfill	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
list.name	TEXT	R/O	StringRange R/O
list.number	INTEGER	R/O	IntegerRange R/O (0 .. 0)
buffer.name	TEXT	R/O	StringRange R/O
savedlist.buf	TEXT	R/O	StringRange R/O
savedlist.name	CHOICE	NOPRESET	EnumRange (?)
save.name	TEXT	NOPRESET	StringRange
savedlist.curr	TEXT	NOPRESET	StringRange
tirg.save	BOOLEAN	NOPRESET	BooleanRange (true false)
tirg.recall	BOOLEAN	NOPRESET	BooleanRange (true false)
tirg.delete	BOOLEAN	NOPRESET	BooleanRange (true false)
internal.cwtest	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alttest	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.1	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.2	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.3	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.collector	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.modemType	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tx Clock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Rx Clock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Rx Demod Unlocked)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Tx Stream)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Rx Stream)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Rx DVBS2 Decoder)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Tx Drop Mux)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Rx Datarate)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Tx Datarate)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx Power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Rx Deframer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tx Mod)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Tx Terr)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Tx IP)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

9.677 Teledyne-QMultiFlex

This page describes the device driver and the device window for the Teledyne-QMuliFlex-400 device. It supports modem functionalities with RF rx and tx interface as well as four ethernet interfaces for traffic. For receiving, the modem utilises 16 separate demodulators. This driver only controls the RF functionalities of the modem. Source and destination IP addresses have to be set manually in the web based interface.

Known issues

When setting or changing the modulation or reloading a stored configuration, the device itself takes up to several seconds to process these changes and is unable. Communication faults that last a few seconds are expected when performing some changes due to the nature of the QMuliFlex-400 device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Lets the user set the mesh topology as well as TX/RX-frequencies and power.
- --- Contains the parameters for the TX modulator such as data-/symbolrate, modulation, fec etc.
- --- Contains similar parameters but for demodulation and dusplayes power values, lock status and RF receive status e.g. eb/n0 or power level.
- --- Displays the current bit- and datarate, dropped packets and error packets for tx and each of the 16 rx channels.
- --- Enables the saving and recalling internally stored device configurations.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Defines the delay the driver uses when it reads back parameters

recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

- pupPassword --- The password for using PUP Commands is to be set here as a string. The internal PUP Command "console" is used by the driver to save configurations in the device internally, delete them or acces them to be applicated.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.modeltype	TEXT	R/O	StringRange R/O
info.serialnumber	TEXT	R/O	StringRange R/O
info.swrevision	TEXT	R/O	StringRange R/O
info.fwrevision	TEXT	R/O	StringRange R/O
info.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.pupPassword	TEXT	SAVE SETUP	StringRange
network.topology	CHOICE	NOPRESET	EnumRange (MESH STAR)
tx.mod.rateControl	CHOICE		EnumRange (DATA RATE SYMBOLRATE)
tx.mod.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.carrierBandwidth3dB	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.carrierBandwidth30dB	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.fecType	CHOICE	NOPRESET	EnumRange (DVBS2 DVBS2X)
tx.mod.fecFramesize	CHOICE	NOPRESET	EnumRange (SHORT NORMAL)
tx.mod.pilots	CHOICE	NOPRESET	EnumRange (ON OFF)
tx.mod.type	CHOICE		EnumRange (QPSK 8PSK 16APSK 32APSK 64APSK 8APSK)
tx.mod.fec	CHOICE	NOPRESET	EnumRange (1/2 2/3 3/4 1/4 1/3 2/5 3/5 4/5 5/6 8/9 9/10 11/45 4/15 13/45 14/45 9/20 7/15 8/15 11/20 26/45 28/45 23/36 25/36 32/45 13/18 11/15 7/9 77/90 1/2-L 8/15-L 5/9-L 26/45-L 3/5-L 2/3-L 32/45-L)
tx.band	CHOICE	NOPRESET	EnumRange (L-BAND IF)
tx.on	CHOICE	NOPRESET	EnumRange (ON OFF RX ENABLED)
tx.power	FLOAT		DoubleRange (-45.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.mod.spectrumInvert	CHOICE	NOPRESET	EnumRange (ON OFF)
tx.mod.rollOff	CHOICE		EnumRange (5% 10% 15% 20% 25% 35%)
rx.demod.band	CHOICE	NOPRESET	EnumRange (L-BAND IF)
rx.demod.spectrumInversion	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.1.enable	CHOICE	NOPRESET	EnumRange (ON OFF)

name	*type*	*flags*	*range*
rx.demod.1.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.1.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.1.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.1.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.1.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.1.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.1.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.1.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.2.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.2.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.2.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.2.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.2.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.2.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.2.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.2.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.2.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.3.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.3.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.3.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)

name	*type*	*flags*	*range*
rx.demod.3.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.3.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.3.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.3.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.3.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.3.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.4.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.4.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.4.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.4.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.4.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.4.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.4.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.4.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.4.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.5.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.5.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.5.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.5.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.5.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.5.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
rx.demod.5.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.5.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.5.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.6.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.6.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.6.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.6.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.6.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.6.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.6.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.6.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.6.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.7.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.7.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.7.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.7.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.7.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.7.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.7.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.7.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.7.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)

name	*type*	*flags*	*range*
rx.demod.8.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.8.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.8.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.8.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.8.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.8.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.8.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.8.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.8.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.9.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.9.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.9.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.9.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.9.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.9.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.9.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.9.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.9.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.10.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.10.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.10.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)

name	*type*	*flags*	*range*
rx.demod.10.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.10.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.10.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.10.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.10.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.10.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.11.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.11.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.11.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.11.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.11.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.11.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.11.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.11.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.11.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.12.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.12.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.12.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.12.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.12.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.12.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
rx.demod.12.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.12.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.12.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.13.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.13.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.13.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.13.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.13.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.13.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.13.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.13.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.13.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.14.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.14.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.14.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.14.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.14.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.14.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.14.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.14.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.14.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)

name	*type*	*flags*	*range*
rx.demod.15.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.15.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.15.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.15.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.15.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.15.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.15.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.15.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.15.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
rx.demod.16.enable	CHOICE	NOPRESET	EnumRange (ON OFF)
rx.demod.16.freq	FLOAT		DoubleRange (50.0000 .. 2450.0000)
rx.demod.16.symbolrate	FLOAT		DoubleRange (0.018000 .. 40.000000)
rx.demod.16.esno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.16.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.16.pwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.16.freqOffset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rx.demod.16.locked	BOOLEAN	R/O	BooleanRange R/O (true false)
rx.demod.16.rolloff	CHOICE	R/O	EnumRange R/O (5% 10% 15% 20% 25% 35%)
tx.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
tx.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
tx.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.1.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.1.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.1.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.1.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.2.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.2.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.2.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.2.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.3.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.3.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.3.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.3.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.4.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.4.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.4.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.4.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.5.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.5.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.5.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.5.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.6.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.6.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
rx.6.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.6.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.7.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.7.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.7.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.7.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.8.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.8.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.8.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.8.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.9.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.9.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.9.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.9.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.10.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.10.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.10.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.10.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.11.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.11.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.11.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.11.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.12.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
rx.12.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.12.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.12.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.13.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.13.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.13.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.13.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.14.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.14.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.14.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.14.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.15.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.15.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.15.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.15.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.16.data.bitrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.16.data.packetrate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.16.data.errpckts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rx.16.data.droppedpackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
save.name	TEXT	NOPRESET	StringRange
savedlist.curr	TEXT	NOPRESET	StringRange
tirg.save	BOOLEAN	NOPRESET	BooleanRange (true false)
tirg.recall	BOOLEAN	NOPRESET	BooleanRange (true false)
tirg.delete	BOOLEAN	NOPRESET	BooleanRange (true false)

name	*type*	*flags*	*range*
list.name	TEXT	R/O	StringRange R/O
list.number	INTEGER	R/O	IntegerRange R/O (0 .. 0)
buffer.name	TEXT	R/O	StringRange R/O
savedlist.buf	TEXT	R/O	StringRange R/O
savedlist.name	CHOICE	R/O	EnumRange R/O (?)
internal.dem.1.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.2.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.3.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.4.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.5.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.6.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.7.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.8.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.9.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.10.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.11.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.12.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.13.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.14.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.15.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.dem.16.locked	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.binstat.01	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.02	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.03	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.04	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.05	HEX	R/O	HexRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.binstat.06	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.07	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.08	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.09	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.10	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.11	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.12	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.13	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.14	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.15	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.16	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.17	HEX	R/O	HexRange R/O (0 .. 0)
internal.binstat.18	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.1	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.2	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.3	HEX	R/O	HexRange R/O (0 .. 0)
internal.sumfault.collector	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tx Clock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Rx Clock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Internal Alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Rx Demod Unlocked)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Tx Stream)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Rx DVBS2 Decoder)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Tx Drop Mux)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Rx Datarate)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Tx Datarate)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Rx Power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Rx Deframer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Tx Mod)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Tx Terr)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Tx IP)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Summary Alarm)

9.678 Temperature-Converter

This page describes the logical temperature converter device driver.

It converts temperatures between kelvin, celsius and fahrenheit.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The converted value.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which to set up the temperature converter.

parameter	*description*
input.Id	Defines the parameter id of the temperature value to be converted
from	Defines the temperature unit of the original value
to	Defines the temperature unit of the desired value to be calculated

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.input.Id	TEXT	SAVE SETUP	StringRange
config.from	CHOICE	SAVE SETUP	EnumRange (CELSIUS FAHRENHEIT KELVIN)
config.to	CHOICE	SAVE SETUP	EnumRange (CELSIUS FAHRENHEIT KELVIN)
input	FLOAT		DoubleRange (0.00 .. 0.00)
value	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.t	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

9.679 Teracue-TAC

This page describes the device driver and the device window for the Teracue TAC transport stream router / transcoder. The driver only supports the routing function of the device, not the transcoding of transport streams.

For the operator, the driver presents the device as a switch matrix for transport streams with a maximum size of 64 inputs x 64 outputs. The only reports inputs / outputs which have been configured in the device's web interface.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Port switching by name 1-32.
- abc --- Port switching by name 33-64.
- 123 --- Port switching by number 1-32.
- 123 --- Port switching by number 33-64.
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The HTTP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp loginName password*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number. The port number may be omitted, the driver uses the standard port 80 in this case.
  - *loginName* --- The login name required to login at the device.
  - *password* --- The password required to login at the device.

Input / Output configuration

The device driver only supports to set and remove crosspoints between inputs and outputs. It does not permit to add or remove inputs/outputs or to change the properties of an input or output. This must be done with the device's Web GUI.

For your convenience, the device window provides a shortcut button to show the device's Web GUI in a browser window. After the device configuration has been changed, the "re-read configuration" button should be clicked to tell the device driver to re-read the complete configuration of the device.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
output.01.present	CHOICE	R/O	EnumRange R/O (0 1)
output.02.present	CHOICE	R/O	EnumRange R/O (0 1)
output.03.present	CHOICE	R/O	EnumRange R/O (0 1)
output.04.present	CHOICE	R/O	EnumRange R/O (0 1)
output.05.present	CHOICE	R/O	EnumRange R/O (0 1)
output.06.present	CHOICE	R/O	EnumRange R/O (0 1)
output.07.present	CHOICE	R/O	EnumRange R/O (0 1)
output.08.present	CHOICE	R/O	EnumRange R/O (0 1)
output.09.present	CHOICE	R/O	EnumRange R/O (0 1)
output.10.present	CHOICE	R/O	EnumRange R/O (0 1)
output.11.present	CHOICE	R/O	EnumRange R/O (0 1)
output.12.present	CHOICE	R/O	EnumRange R/O (0 1)
output.13.present	CHOICE	R/O	EnumRange R/O (0 1)
output.14.present	CHOICE	R/O	EnumRange R/O (0 1)
output.15.present	CHOICE	R/O	EnumRange R/O (0 1)
output.16.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.17.present	CHOICE	R/O	EnumRange R/O (0 1)
output.18.present	CHOICE	R/O	EnumRange R/O (0 1)
output.19.present	CHOICE	R/O	EnumRange R/O (0 1)
output.20.present	CHOICE	R/O	EnumRange R/O (0 1)
output.21.present	CHOICE	R/O	EnumRange R/O (0 1)
output.22.present	CHOICE	R/O	EnumRange R/O (0 1)
output.23.present	CHOICE	R/O	EnumRange R/O (0 1)
output.24.present	CHOICE	R/O	EnumRange R/O (0 1)
output.25.present	CHOICE	R/O	EnumRange R/O (0 1)
output.26.present	CHOICE	R/O	EnumRange R/O (0 1)
output.27.present	CHOICE	R/O	EnumRange R/O (0 1)
output.28.present	CHOICE	R/O	EnumRange R/O (0 1)
output.29.present	CHOICE	R/O	EnumRange R/O (0 1)
output.30.present	CHOICE	R/O	EnumRange R/O (0 1)
output.31.present	CHOICE	R/O	EnumRange R/O (0 1)
output.32.present	CHOICE	R/O	EnumRange R/O (0 1)
output.33.present	CHOICE	R/O	EnumRange R/O (0 1)
output.34.present	CHOICE	R/O	EnumRange R/O (0 1)
output.35.present	CHOICE	R/O	EnumRange R/O (0 1)
output.36.present	CHOICE	R/O	EnumRange R/O (0 1)
output.37.present	CHOICE	R/O	EnumRange R/O (0 1)
output.38.present	CHOICE	R/O	EnumRange R/O (0 1)
output.39.present	CHOICE	R/O	EnumRange R/O (0 1)
output.40.present	CHOICE	R/O	EnumRange R/O (0 1)
output.41.present	CHOICE	R/O	EnumRange R/O (0 1)
output.42.present	CHOICE	R/O	EnumRange R/O (0 1)
output.43.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
output.44.present	CHOICE	R/O	EnumRange R/O (0 1)
output.45.present	CHOICE	R/O	EnumRange R/O (0 1)
output.46.present	CHOICE	R/O	EnumRange R/O (0 1)
output.47.present	CHOICE	R/O	EnumRange R/O (0 1)
output.48.present	CHOICE	R/O	EnumRange R/O (0 1)
output.49.present	CHOICE	R/O	EnumRange R/O (0 1)
output.50.present	CHOICE	R/O	EnumRange R/O (0 1)
output.51.present	CHOICE	R/O	EnumRange R/O (0 1)
output.52.present	CHOICE	R/O	EnumRange R/O (0 1)
output.53.present	CHOICE	R/O	EnumRange R/O (0 1)
output.54.present	CHOICE	R/O	EnumRange R/O (0 1)
output.55.present	CHOICE	R/O	EnumRange R/O (0 1)
output.56.present	CHOICE	R/O	EnumRange R/O (0 1)
output.57.present	CHOICE	R/O	EnumRange R/O (0 1)
output.58.present	CHOICE	R/O	EnumRange R/O (0 1)
output.59.present	CHOICE	R/O	EnumRange R/O (0 1)
output.60.present	CHOICE	R/O	EnumRange R/O (0 1)
output.61.present	CHOICE	R/O	EnumRange R/O (0 1)
output.62.present	CHOICE	R/O	EnumRange R/O (0 1)
output.63.present	CHOICE	R/O	EnumRange R/O (0 1)
output.64.present	CHOICE	R/O	EnumRange R/O (0 1)
output.65.present	CHOICE	R/O	EnumRange R/O (0 1)
output.66.present	CHOICE	R/O	EnumRange R/O (0 1)
output.67.present	CHOICE	R/O	EnumRange R/O (0 1)
output.68.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.01.present	CHOICE	R/O	EnumRange R/O (0 1)
input.02.present	CHOICE	R/O	EnumRange R/O (0 1)
input.03.present	CHOICE	R/O	EnumRange R/O (0 1)
input.04.present	CHOICE	R/O	EnumRange R/O (0 1)
input.05.present	CHOICE	R/O	EnumRange R/O (0 1)
input.06.present	CHOICE	R/O	EnumRange R/O (0 1)
input.07.present	CHOICE	R/O	EnumRange R/O (0 1)
input.08.present	CHOICE	R/O	EnumRange R/O (0 1)
input.09.present	CHOICE	R/O	EnumRange R/O (0 1)
input.10.present	CHOICE	R/O	EnumRange R/O (0 1)
input.11.present	CHOICE	R/O	EnumRange R/O (0 1)
input.12.present	CHOICE	R/O	EnumRange R/O (0 1)
input.13.present	CHOICE	R/O	EnumRange R/O (0 1)
input.14.present	CHOICE	R/O	EnumRange R/O (0 1)
input.15.present	CHOICE	R/O	EnumRange R/O (0 1)
input.16.present	CHOICE	R/O	EnumRange R/O (0 1)
input.17.present	CHOICE	R/O	EnumRange R/O (0 1)
input.18.present	CHOICE	R/O	EnumRange R/O (0 1)
input.19.present	CHOICE	R/O	EnumRange R/O (0 1)
input.20.present	CHOICE	R/O	EnumRange R/O (0 1)
input.21.present	CHOICE	R/O	EnumRange R/O (0 1)
input.22.present	CHOICE	R/O	EnumRange R/O (0 1)
input.23.present	CHOICE	R/O	EnumRange R/O (0 1)
input.24.present	CHOICE	R/O	EnumRange R/O (0 1)
input.25.present	CHOICE	R/O	EnumRange R/O (0 1)
input.26.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.27.present	CHOICE	R/O	EnumRange R/O (0 1)
input.28.present	CHOICE	R/O	EnumRange R/O (0 1)
input.29.present	CHOICE	R/O	EnumRange R/O (0 1)
input.30.present	CHOICE	R/O	EnumRange R/O (0 1)
input.31.present	CHOICE	R/O	EnumRange R/O (0 1)
input.32.present	CHOICE	R/O	EnumRange R/O (0 1)
input.33.present	CHOICE	R/O	EnumRange R/O (0 1)
input.34.present	CHOICE	R/O	EnumRange R/O (0 1)
input.35.present	CHOICE	R/O	EnumRange R/O (0 1)
input.36.present	CHOICE	R/O	EnumRange R/O (0 1)
input.37.present	CHOICE	R/O	EnumRange R/O (0 1)
input.38.present	CHOICE	R/O	EnumRange R/O (0 1)
input.39.present	CHOICE	R/O	EnumRange R/O (0 1)
input.40.present	CHOICE	R/O	EnumRange R/O (0 1)
input.41.present	CHOICE	R/O	EnumRange R/O (0 1)
input.42.present	CHOICE	R/O	EnumRange R/O (0 1)
input.43.present	CHOICE	R/O	EnumRange R/O (0 1)
input.44.present	CHOICE	R/O	EnumRange R/O (0 1)
input.45.present	CHOICE	R/O	EnumRange R/O (0 1)
input.46.present	CHOICE	R/O	EnumRange R/O (0 1)
input.47.present	CHOICE	R/O	EnumRange R/O (0 1)
input.48.present	CHOICE	R/O	EnumRange R/O (0 1)
input.49.present	CHOICE	R/O	EnumRange R/O (0 1)
input.50.present	CHOICE	R/O	EnumRange R/O (0 1)
input.51.present	CHOICE	R/O	EnumRange R/O (0 1)
input.52.present	CHOICE	R/O	EnumRange R/O (0 1)

name	*type*	*flags*	*range*
input.53.present	CHOICE	R/O	EnumRange R/O (0 1)
input.54.present	CHOICE	R/O	EnumRange R/O (0 1)
input.55.present	CHOICE	R/O	EnumRange R/O (0 1)
input.56.present	CHOICE	R/O	EnumRange R/O (0 1)
input.57.present	CHOICE	R/O	EnumRange R/O (0 1)
input.58.present	CHOICE	R/O	EnumRange R/O (0 1)
input.59.present	CHOICE	R/O	EnumRange R/O (0 1)
input.60.present	CHOICE	R/O	EnumRange R/O (0 1)
input.61.present	CHOICE	R/O	EnumRange R/O (0 1)
input.62.present	CHOICE	R/O	EnumRange R/O (0 1)
input.63.present	CHOICE	R/O	EnumRange R/O (0 1)
input.64.present	CHOICE	R/O	EnumRange R/O (0 1)
output.01.name	TEXT	R/O	StringRange R/O
output.02.name	TEXT	R/O	StringRange R/O
output.03.name	TEXT	R/O	StringRange R/O
output.04.name	TEXT	R/O	StringRange R/O
output.05.name	TEXT	R/O	StringRange R/O
output.06.name	TEXT	R/O	StringRange R/O
output.07.name	TEXT	R/O	StringRange R/O
output.08.name	TEXT	R/O	StringRange R/O
output.09.name	TEXT	R/O	StringRange R/O
output.10.name	TEXT	R/O	StringRange R/O
output.11.name	TEXT	R/O	StringRange R/O
output.12.name	TEXT	R/O	StringRange R/O
output.13.name	TEXT	R/O	StringRange R/O
output.14.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.15.name	TEXT	R/O	StringRange R/O
output.16.name	TEXT	R/O	StringRange R/O
output.17.name	TEXT	R/O	StringRange R/O
output.18.name	TEXT	R/O	StringRange R/O
output.19.name	TEXT	R/O	StringRange R/O
output.20.name	TEXT	R/O	StringRange R/O
output.21.name	TEXT	R/O	StringRange R/O
output.22.name	TEXT	R/O	StringRange R/O
output.23.name	TEXT	R/O	StringRange R/O
output.24.name	TEXT	R/O	StringRange R/O
output.25.name	TEXT	R/O	StringRange R/O
output.26.name	TEXT	R/O	StringRange R/O
output.27.name	TEXT	R/O	StringRange R/O
output.28.name	TEXT	R/O	StringRange R/O
output.29.name	TEXT	R/O	StringRange R/O
output.30.name	TEXT	R/O	StringRange R/O
output.31.name	TEXT	R/O	StringRange R/O
output.32.name	TEXT	R/O	StringRange R/O
output.33.name	TEXT	R/O	StringRange R/O
output.34.name	TEXT	R/O	StringRange R/O
output.35.name	TEXT	R/O	StringRange R/O
output.36.name	TEXT	R/O	StringRange R/O
output.37.name	TEXT	R/O	StringRange R/O
output.38.name	TEXT	R/O	StringRange R/O
output.39.name	TEXT	R/O	StringRange R/O
output.40.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.41.name	TEXT	R/O	StringRange R/O
output.42.name	TEXT	R/O	StringRange R/O
output.43.name	TEXT	R/O	StringRange R/O
output.44.name	TEXT	R/O	StringRange R/O
output.45.name	TEXT	R/O	StringRange R/O
output.46.name	TEXT	R/O	StringRange R/O
output.47.name	TEXT	R/O	StringRange R/O
output.48.name	TEXT	R/O	StringRange R/O
output.49.name	TEXT	R/O	StringRange R/O
output.50.name	TEXT	R/O	StringRange R/O
output.51.name	TEXT	R/O	StringRange R/O
output.52.name	TEXT	R/O	StringRange R/O
output.53.name	TEXT	R/O	StringRange R/O
output.54.name	TEXT	R/O	StringRange R/O
output.55.name	TEXT	R/O	StringRange R/O
output.56.name	TEXT	R/O	StringRange R/O
output.57.name	TEXT	R/O	StringRange R/O
output.58.name	TEXT	R/O	StringRange R/O
output.59.name	TEXT	R/O	StringRange R/O
output.60.name	TEXT	R/O	StringRange R/O
output.61.name	TEXT	R/O	StringRange R/O
output.62.name	TEXT	R/O	StringRange R/O
output.63.name	TEXT	R/O	StringRange R/O
output.64.name	TEXT	R/O	StringRange R/O
output.65.name	TEXT	R/O	StringRange R/O
output.66.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.67.name	TEXT	R/O	StringRange R/O
output.68.name	TEXT	R/O	StringRange R/O
input.01.name	TEXT	R/O	StringRange R/O
input.02.name	TEXT	R/O	StringRange R/O
input.03.name	TEXT	R/O	StringRange R/O
input.04.name	TEXT	R/O	StringRange R/O
input.05.name	TEXT	R/O	StringRange R/O
input.06.name	TEXT	R/O	StringRange R/O
input.07.name	TEXT	R/O	StringRange R/O
input.08.name	TEXT	R/O	StringRange R/O
input.09.name	TEXT	R/O	StringRange R/O
input.10.name	TEXT	R/O	StringRange R/O
input.11.name	TEXT	R/O	StringRange R/O
input.12.name	TEXT	R/O	StringRange R/O
input.13.name	TEXT	R/O	StringRange R/O
input.14.name	TEXT	R/O	StringRange R/O
input.15.name	TEXT	R/O	StringRange R/O
input.16.name	TEXT	R/O	StringRange R/O
input.17.name	TEXT	R/O	StringRange R/O
input.18.name	TEXT	R/O	StringRange R/O
input.19.name	TEXT	R/O	StringRange R/O
input.20.name	TEXT	R/O	StringRange R/O
input.21.name	TEXT	R/O	StringRange R/O
input.22.name	TEXT	R/O	StringRange R/O
input.23.name	TEXT	R/O	StringRange R/O
input.24.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
input.25.name	TEXT	R/O	StringRange R/O
input.26.name	TEXT	R/O	StringRange R/O
input.27.name	TEXT	R/O	StringRange R/O
input.28.name	TEXT	R/O	StringRange R/O
input.29.name	TEXT	R/O	StringRange R/O
input.30.name	TEXT	R/O	StringRange R/O
input.31.name	TEXT	R/O	StringRange R/O
input.32.name	TEXT	R/O	StringRange R/O
input.33.name	TEXT	R/O	StringRange R/O
input.34.name	TEXT	R/O	StringRange R/O
input.35.name	TEXT	R/O	StringRange R/O
input.36.name	TEXT	R/O	StringRange R/O
input.37.name	TEXT	R/O	StringRange R/O
input.38.name	TEXT	R/O	StringRange R/O
input.39.name	TEXT	R/O	StringRange R/O
input.40.name	TEXT	R/O	StringRange R/O
input.41.name	TEXT	R/O	StringRange R/O
input.42.name	TEXT	R/O	StringRange R/O
input.43.name	TEXT	R/O	StringRange R/O
input.44.name	TEXT	R/O	StringRange R/O
input.45.name	TEXT	R/O	StringRange R/O
input.46.name	TEXT	R/O	StringRange R/O
input.47.name	TEXT	R/O	StringRange R/O
input.48.name	TEXT	R/O	StringRange R/O
input.49.name	TEXT	R/O	StringRange R/O
input.50.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
input.51.name	TEXT	R/O	StringRange R/O
input.52.name	TEXT	R/O	StringRange R/O
input.53.name	TEXT	R/O	StringRange R/O
input.54.name	TEXT	R/O	StringRange R/O
input.55.name	TEXT	R/O	StringRange R/O
input.56.name	TEXT	R/O	StringRange R/O
input.57.name	TEXT	R/O	StringRange R/O
input.58.name	TEXT	R/O	StringRange R/O
input.59.name	TEXT	R/O	StringRange R/O
input.60.name	TEXT	R/O	StringRange R/O
input.61.name	TEXT	R/O	StringRange R/O
input.62.name	TEXT	R/O	StringRange R/O
input.63.name	TEXT	R/O	StringRange R/O
input.64.name	TEXT	R/O	StringRange R/O
output.01	CHOICE		EnumRange (OFF)
output.02	CHOICE		EnumRange (OFF)
output.03	CHOICE		EnumRange (OFF)
output.04	CHOICE		EnumRange (OFF)
output.05	CHOICE		EnumRange (OFF)
output.06	CHOICE		EnumRange (OFF)
output.07	CHOICE		EnumRange (OFF)
output.08	CHOICE		EnumRange (OFF)
output.09	CHOICE		EnumRange (OFF)
output.10	CHOICE		EnumRange (OFF)
output.11	CHOICE		EnumRange (OFF)
output.12	CHOICE		EnumRange (OFF)
output.13	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.14	CHOICE		EnumRange (OFF)
output.15	CHOICE		EnumRange (OFF)
output.16	CHOICE		EnumRange (OFF)
output.17	CHOICE		EnumRange (OFF)
output.18	CHOICE		EnumRange (OFF)
output.19	CHOICE		EnumRange (OFF)
output.20	CHOICE		EnumRange (OFF)
output.21	CHOICE		EnumRange (OFF)
output.22	CHOICE		EnumRange (OFF)
output.23	CHOICE		EnumRange (OFF)
output.24	CHOICE		EnumRange (OFF)
output.25	CHOICE		EnumRange (OFF)
output.26	CHOICE		EnumRange (OFF)
output.27	CHOICE		EnumRange (OFF)
output.28	CHOICE		EnumRange (OFF)
output.29	CHOICE		EnumRange (OFF)
output.30	CHOICE		EnumRange (OFF)
output.31	CHOICE		EnumRange (OFF)
output.32	CHOICE		EnumRange (OFF)
output.33	CHOICE		EnumRange (OFF)
output.34	CHOICE		EnumRange (OFF)
output.35	CHOICE		EnumRange (OFF)
output.36	CHOICE		EnumRange (OFF)
output.37	CHOICE		EnumRange (OFF)
output.38	CHOICE		EnumRange (OFF)
output.39	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.40	CHOICE		EnumRange (OFF)
output.41	CHOICE		EnumRange (OFF)
output.42	CHOICE		EnumRange (OFF)
output.43	CHOICE		EnumRange (OFF)
output.44	CHOICE		EnumRange (OFF)
output.45	CHOICE		EnumRange (OFF)
output.46	CHOICE		EnumRange (OFF)
output.47	CHOICE		EnumRange (OFF)
output.48	CHOICE		EnumRange (OFF)
output.49	CHOICE		EnumRange (OFF)
output.50	CHOICE		EnumRange (OFF)
output.51	CHOICE		EnumRange (OFF)
output.52	CHOICE		EnumRange (OFF)
output.53	CHOICE		EnumRange (OFF)
output.54	CHOICE		EnumRange (OFF)
output.55	CHOICE		EnumRange (OFF)
output.56	CHOICE		EnumRange (OFF)
output.57	CHOICE		EnumRange (OFF)
output.58	CHOICE		EnumRange (OFF)
output.59	CHOICE		EnumRange (OFF)
output.60	CHOICE		EnumRange (OFF)
output.61	CHOICE		EnumRange (OFF)
output.62	CHOICE		EnumRange (OFF)
output.63	CHOICE		EnumRange (OFF)
output.64	CHOICE		EnumRange (OFF)

name	*type*	*flags*	*range*
output.65	CHOICE		EnumRange (OFF)
output.66	CHOICE		EnumRange (OFF)
output.67	CHOICE		EnumRange (OFF)
output.68	CHOICE		EnumRange (OFF)
byname.01	CHOICE	NOPRESET	EnumRange (OFF)
byname.02	CHOICE	NOPRESET	EnumRange (OFF)
byname.03	CHOICE	NOPRESET	EnumRange (OFF)
byname.04	CHOICE	NOPRESET	EnumRange (OFF)
byname.05	CHOICE	NOPRESET	EnumRange (OFF)
byname.06	CHOICE	NOPRESET	EnumRange (OFF)
byname.07	CHOICE	NOPRESET	EnumRange (OFF)
byname.08	CHOICE	NOPRESET	EnumRange (OFF)
byname.09	CHOICE	NOPRESET	EnumRange (OFF)
byname.10	CHOICE	NOPRESET	EnumRange (OFF)
byname.11	CHOICE	NOPRESET	EnumRange (OFF)
byname.12	CHOICE	NOPRESET	EnumRange (OFF)
byname.13	CHOICE	NOPRESET	EnumRange (OFF)
byname.14	CHOICE	NOPRESET	EnumRange (OFF)
byname.15	CHOICE	NOPRESET	EnumRange (OFF)
byname.16	CHOICE	NOPRESET	EnumRange (OFF)
byname.17	CHOICE	NOPRESET	EnumRange (OFF)
byname.18	CHOICE	NOPRESET	EnumRange (OFF)
byname.19	CHOICE	NOPRESET	EnumRange (OFF)
byname.20	CHOICE	NOPRESET	EnumRange (OFF)
byname.21	CHOICE	NOPRESET	EnumRange (OFF)
byname.22	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.23	CHOICE	NOPRESET	EnumRange (OFF)
byname.24	CHOICE	NOPRESET	EnumRange (OFF)
byname.25	CHOICE	NOPRESET	EnumRange (OFF)
byname.26	CHOICE	NOPRESET	EnumRange (OFF)
byname.27	CHOICE	NOPRESET	EnumRange (OFF)
byname.28	CHOICE	NOPRESET	EnumRange (OFF)
byname.29	CHOICE	NOPRESET	EnumRange (OFF)
byname.30	CHOICE	NOPRESET	EnumRange (OFF)
byname.31	CHOICE	NOPRESET	EnumRange (OFF)
byname.32	CHOICE	NOPRESET	EnumRange (OFF)
byname.33	CHOICE	NOPRESET	EnumRange (OFF)
byname.34	CHOICE	NOPRESET	EnumRange (OFF)
byname.35	CHOICE	NOPRESET	EnumRange (OFF)
byname.36	CHOICE	NOPRESET	EnumRange (OFF)
byname.37	CHOICE	NOPRESET	EnumRange (OFF)
byname.38	CHOICE	NOPRESET	EnumRange (OFF)
byname.39	CHOICE	NOPRESET	EnumRange (OFF)
byname.40	CHOICE	NOPRESET	EnumRange (OFF)
byname.41	CHOICE	NOPRESET	EnumRange (OFF)
byname.42	CHOICE	NOPRESET	EnumRange (OFF)
byname.43	CHOICE	NOPRESET	EnumRange (OFF)
byname.44	CHOICE	NOPRESET	EnumRange (OFF)
byname.45	CHOICE	NOPRESET	EnumRange (OFF)
byname.46	CHOICE	NOPRESET	EnumRange (OFF)
byname.47	CHOICE	NOPRESET	EnumRange (OFF)
byname.48	CHOICE	NOPRESET	EnumRange (OFF)

name	*type*	*flags*	*range*
byname.49	CHOICE	NOPRESET	EnumRange (OFF)
byname.50	CHOICE	NOPRESET	EnumRange (OFF)
byname.51	CHOICE	NOPRESET	EnumRange (OFF)
byname.52	CHOICE	NOPRESET	EnumRange (OFF)
byname.53	CHOICE	NOPRESET	EnumRange (OFF)
byname.54	CHOICE	NOPRESET	EnumRange (OFF)
byname.55	CHOICE	NOPRESET	EnumRange (OFF)
byname.56	CHOICE	NOPRESET	EnumRange (OFF)
byname.57	CHOICE	NOPRESET	EnumRange (OFF)
byname.58	CHOICE	NOPRESET	EnumRange (OFF)
byname.59	CHOICE	NOPRESET	EnumRange (OFF)
byname.60	CHOICE	NOPRESET	EnumRange (OFF)
byname.61	CHOICE	NOPRESET	EnumRange (OFF)
byname.62	CHOICE	NOPRESET	EnumRange (OFF)
byname.63	CHOICE	NOPRESET	EnumRange (OFF)
byname.64	CHOICE	NOPRESET	EnumRange (OFF)
byname.65	CHOICE	NOPRESET	EnumRange (OFF)
byname.66	CHOICE	NOPRESET	EnumRange (OFF)
byname.67	CHOICE	NOPRESET	EnumRange (OFF)
byname.68	CHOICE	NOPRESET	EnumRange (OFF)
reset	TEXT	NOPRESET	StringRange

9.680 Terrasat-IBUC-SNMP

This page describes the device driver and the device window for the Terrasat Ku-Band and C-Band L-Band Blockupconverter (IBUC) and supports the built-in 1+1 TX redundancy configuration.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- --- Meter readings
- --- Protection switch parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	***description***
address	The device driver uses the terminal protocol to talk with the transceiver, hence there is no device addressing. Leave this field empty.
externalLED	This enables or disables the LED on the chassis.
outputHighLimit	Output high power level trehshold
outputLowLimit	Output low power level threshold
inputHighLimit	Input high power level threshold
inputLowLimit	Input low power level threshold
password	This is the passwort which is used for login
txPowerupDelay	Sets the delay for tx on at power up
txPowerupState	If set to 'OFF', the IBUC starts with RF OFF when it is powered up. The 'LAST STATE' setting lets the transceiver return to the state when it was switched off

Ethernet connection

If you want to use the TCP/IP M&C interface you have to configure the IP address of the IBUC first. Please refer to the Terrasat IBUC User Manual. If you want to use the telnet M&C interface or the serial M&C interface please use the driver Terrasat-IBUC

SNMP configuration

The following commands (via telnet or serial interface) configure SNMP operation on the IBUC:

command	*default*	*description*
CIT	0	This command enables or disables the SNMP trap capabilities of the IBUC. CIT=1 enables traps. SNMP must be enabled to generate traps(CIS=1). CIT=0 disables traps. CIT? returns the current setting of traps.
CIS	0	This command enables or disables the SNMP functionality of the IBUC. CIS=1 enables SNMPV1. CIS=0 disables SNMPV1. CIS? returns the current setting of SNMP functionality.
CIC	public,private,trap	sets the SNMP passwords for the IBUC. CIC=<public>,<private>,<trap> where public is the public password, private is the private password, and trap is the trap password. All three passwords are required when issuing this command. Each password cannot exceed 16 characters. Passwords may not be read for security purposes.

1+1 Redundancy / Protection switching

If you have a 1+1 IBUC TX redundancy you have to configure both IBUC's first via web browser interface. Then you can configure two devices (e.g. BUC-A and BUC-B) within the sat-nms M&C system. The active unit (that one which is not switched to dummy load) is automatically the master unit. You can only send commands to the master. The slave unit will not accept any commands.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.outputHighLimit	FLOAT	SETUP	DoubleRange (10.00 .. 60.00)
config.outputLowLimit	FLOAT	SETUP	DoubleRange (10.00 .. 60.00)
config.inputHighLimit	FLOAT	SETUP	DoubleRange (-60.00 .. -15.00)
config.inputLowLimit	FLOAT	SETUP	DoubleRange (-60.00 .. -15.00)
config.txPowerupDelay	INTEGER	SETUP	IntegerRange (0 .. 500)
config.txPowerupState	CHOICE	SETUP	EnumRange (LAST STATE OFF)
config.highTempShutdown	CHOICE	SETUP	EnumRange (ON OFF)
config.txBand	CHOICE	SETUP	EnumRange (SINGLE BAND COMMERCIAL-RF-BAND-1 COMMERCIAL-RF-BAND-2 COMMERCIAL-RF-BAND-3 MILITARY-RF-BAND-1 MILITARY-RF-BAND-2)
config.ifBand	CHOICE	SETUP	EnumRange (IF-BAND-1 IF-BAND-2)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.powerClass	CHOICE	R/O	EnumRange R/O (2 Watt 4 Watt 5 Watt 8 Watt 10 Watt 16 Watt 20 Watt 25 Watt 40 Watt 60 Watt 30 Watt 80 Watt 100 Watt 12 Watt 50 Watt 125 Watt 150 Watt 200 Watt 175 Watt 400 Watt 15 Watt)
info.ipAddress	TEXT	R/O	StringRange R/O
info.otherAddress	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.loFrequency	CHOICE	R/O	EnumRange R/O (Standard C-Band Instat C-Band Extended Ku-Band Standard Ku-Band X-Band 7 Full Ku-Band Extended C-Band Full C-Band Low Ku-Band Palapa C-Band High Ku-Band DBS-Band 1 DBS-Band 2 Ka-Band 1 Ka-Band 2 Ku-Band 4 C-Band 6 Ka-Band 3 Ka-Band 4)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.burstCount	INTEGER		IntegerRange (1 .. 16)
tx.burstThreshold	FLOAT		DoubleRange (0.00 .. 60.00)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 16.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 16.0)
tx.frequency	INTEGER		IntegerRange (0 .. 0)
tx.controlMode	CHOICE		EnumRange (Open loop ALC AGC)
tx.gainTarget	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.levelTarget	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.gainControl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.powerReadMode	CHOICE		EnumRange (CSM (cont.) Burst)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.idc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.pldroVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
protection	CHOICE		EnumRange DIS (MANUAL AUTO)

name	*type*	*flags*	*range*
autoClone	CHOICE		EnumRange DIS (OFF ON)
redundancyMode	CHOICE		EnumRange DIS (reverting non-reverting)
onSwitch	CHOICE		EnumRange DIS (attached as A attached as B)
location	CHOICE		EnumRange DIS (A B)
position	CHOICE		EnumRange DIS (A B)
txSwitch	CHOICE		EnumRange DIS (ANTENNA LOAD)
reset	TEXT	NOPRESET	StringRange
switchedReset	TEXT	NOPRESET	StringRange
internal.redundancy	INTEGER		IntegerRange (0 .. 0)
internal.snmp	TEXT	R/O	StringRange R/O
internal.version	TEXT	R/O	StringRange R/O
internal.model	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (DRO lock alarm)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC not settled)
faults.03	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC target range)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input Level Low)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Input Level High)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output Level Low)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output Level High)
faults.08	ALARM	R/O	AlarmFlagRange R/O (10MHz reference)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC range)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Tx simulated)
faults.12	ALARM	R/O	AlarmFlagRange R/O (VDC range)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IDC range)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Alarm log)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Config data)
faults.16	ALARM	R/O	AlarmFlagRange R/O (WG switch)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Input Overdrive)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Reverse Level high)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.681 Terrasat-IBUC

This page describes the device driver and the device window for the Terrasat Ku-Band and C-Band L-Band Blockupconverter (IBUC).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- This page shows transmit part parameters of the device.
- --- Meter reading
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to make the software talk to the transceiver.

parameter	***description***
address	The device driver uses the terminal protocol to talk with the transceiver, hence there is no device addressing. Leave this field empty.
externalLED	This enables or disables the LED on the chassis.
outputHighLimit	Output high power level threshold
outputLowLimit	Output low power level threshold
inputHighLimit	Input high power level threshold
inputLowLimit	Input low power level threshold
password	This is the passwort which is used for login
txPowerupDelay	Sets the delay for tx on at power up
txPowerupState	If set to 'OFF', the IBUC starts with RF OFF when it is powered up. The 'LAST STATE' setting lets the transceiver return to the state when it was switched off
Serial	Remember to set the line parameters for the serial interface connected to the transceiver. The transceiver either expects *9600,N,8,1* when set to factory settings.

Ethernet connection

If you want to use the TCP/IP M&C interface you have to configure the IP address of the IBUC first. Please refer to the Terrasat IBUC User Manual. The default Port is 23

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.externalLED	CHOICE	SETUP	EnumRange (ON OFF)
config.outputHighLimit	FLOAT	SETUP	DoubleRange (10.00 .. 60.00)
config.outputLowLimit	FLOAT	SETUP	DoubleRange (10.00 .. 60.00)
config.inputHighLimit	FLOAT	SETUP	DoubleRange (-60.00 .. -15.00)
config.inputLowLimit	FLOAT	SETUP	DoubleRange (-60.00 .. -15.00)
config.password	TEXT	SAVE SETUP	StringRange
config.txPowerupDelay	INTEGER	SETUP	IntegerRange (0 .. 500)
config.txPowerupState	CHOICE	SETUP	EnumRange (LAST STATE OFF)
config.highTempShutdown	CHOICE	SETUP	EnumRange (ON OFF)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.powerClass	TEXT	R/O	StringRange R/O
info.ipAddress	TEXT	R/O	StringRange R/O
info.loFrequency	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.burstCount	INTEGER		IntegerRange (1 .. 16)
tx.burstThreshold	FLOAT		DoubleRange (0.0 .. 60.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 16.0)
tx.frequency	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
tx.controlMode	CHOICE		EnumRange (Open Loop ALC AGC)
tx.gainTarget	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.levelTarget	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.gainControl	FLOAT		DoubleRange (0.00 .. 0.00)
tx.powerReadMode	CHOICE		EnumRange (CSM (cont.) Burst)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
meas.vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
meas.idc	FLOAT	R/O	DoubleRange R/O (0.0 .. 12.5)
meas.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.pldroVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 10.0)
internal.faults	HEX	R/O	HexRange R/O (0 .. 0)
internal.misc	HEX	R/O	HexRange R/O (0 .. 0)
internal.login	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC range)
faults.02	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC not settled)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Loss-of-lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Output Level High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output Level Low)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Input Level High)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Input Level Low)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Tx simulated)
faults.10	ALARM	R/O	AlarmFlagRange R/O (10MHz reference)
faults.11	ALARM	R/O	AlarmFlagRange R/O (VDC range)
faults.12	ALARM	R/O	AlarmFlagRange R/O (WG switch)
faults.13	ALARM	R/O	AlarmFlagRange R/O (IDC range)
faults.14	ALARM	R/O	AlarmFlagRange R/O (AGC/ALC target range)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Wrong password)

9.682 Terrasat-RX-Protection

This page describes the device driver and the device window for the Terrasat Receive Protection / Redundancy switching system.

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read

and write community.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.status	TEXT	R/O	StringRange R/O
protection	CHOICE		EnumRange (MANUAL AUTO)
redundancyMode	CHOICE		EnumRange (reverting non-reverting)
location	CHOICE		EnumRange (A B)
position	CHOICE		EnumRange (A B)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.power.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.power.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
internal.fault1	CHOICE	R/O	EnumRange R/O (true false)
internal.fault2	CHOICE	R/O	EnumRange R/O (true false)
internal.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Low input A)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low input B)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Simulated alarm A)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Simulated alarm B)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Low voltage A)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low voltage B)
faults.07	ALARM	R/O	AlarmFlagRange R/O (High voltage A)
faults.08	ALARM	R/O	AlarmFlagRange R/O (High voltage B)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low current A)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Low current B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High current A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (High current B)
faults.13	ALARM	R/O	AlarmFlagRange R/O (10 MHz)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Emergency override)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Switch)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Corrupt log)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Config data)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.683 Threshold

The Threshold logical device monitors one parameter like a signal level and generates

warnings or faults if given threshold are reached.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the current state of the monitored parameter:
  - UNDEF : parameter could not be evaluated
  - OK : no threshold reached
  - WARNING : warningThreshold reached
  - ALARM : alarmThreshold reached.
- --- The fault page
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

- *parameter --- description*
- config.inputId --- The parameter ID of the parameter to monitor.
- config.warningThreshold --- Warning threshold
- config.alarmThreshold --- Alarm threshold
- config.thresholdType --- Direction of threshold monitoring
  - LOW: alarm/warning if value is below thresholds
  - HIGH: alarm/warning if value is above thresholds

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.inputId	TEXT	SAVE SETUP	StringRange
config.warningThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
config.alarmThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0000 .. 0.0000)
config.thresholdType	CHOICE	SAVE SETUP	EnumRange (LOW HIGH)
internal.input	FLOAT		DoubleRange (0.0000 .. 0.0000)
position	CHOICE		EnumRange (UNDEF OK WARNING ALARM)
faults.01	ALARM	R/O	AlarmFlagRange R/O (threshold warning)
faults.02	ALARM	R/O	AlarmFlagRange R/O (threshold alarm)

9.684 TimeLink-Distributor-TMD

This page describes the device driver and the device window for the TimeLink Microsystems distributor TMD series 10MHz/TimeCode distributor devices with redundant input. The driver covers the following devices:

- TMD2312/13 10 MHz redundant low noise distribution unit
- TMD1101 Analogue time code amplifier & distributor

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the device status and controls input selection and operation mode.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The TimeLink devices use a text based protocol over UDP, the 'address' configuration parameter must be set as shown below.

parameter	*description*

parameter	*description*
address	The driver expects the 'address' parameter to be set to the device's IP address, followed by a colon and the port used by the device. Factory default for the port number is 9500, but this may be changed through the Web-GUI of these devices. Example: **192.168.1.245:9500**
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
protection	CHOICE		EnumRange (OFF ON)
location	CHOICE		EnumRange (A B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Channel A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Channel B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU 1 5V)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU 1 12V)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU 1 -12V)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PSU 2 5V)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (PSU 2 12V)
faults.06	ALARM	R/O	AlarmFlagRange R/O (PSU 2 -12V)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.685 TimeLink-Distributor

This page describes the device driver and the device window for the TimeLink Microsystems distributor PPD/IRIG B/10MHz distributor devices with redundant input. The driver covers the following devices:

- SR1101-R IRIG B redundant distribution unit
- SR2214-R-W 10 MHz redundant low noise distribution unit
- SR1132-R PPS redundant distribution unit

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the device status and controls input selection and operation mode.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The TimeLink devices use a text based protocol over UDP, the 'address' configuration parameter must be set as shown below.

parameter	*description*
address	The driver expects the 'address' parameter to be set to the device's IP address, followed by a colon and the port number used by the device. Factory default for the port number is 9500, but this may be changed through the Web-GUI of these devices. Example: **192.168.1.245:9500**

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
protection	CHOICE		EnumRange (OFF ON)
location	CHOICE		EnumRange (A B)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Channel A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Channel B)

9.686 TimeLink-GNSS-Receiver-TMG3400

This page describes the monitoring only device driver and the device window for the TimeLink GNSS Frequency and Time Generator.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The GNSS related parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The unit do not need any configuration parameters except SNMP settings

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.identifier	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
utcTime	TEXT	R/O	StringRange R/O
state.frequency	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.oscillator	TEXT	R/O	StringRange R/O
oscillator.dac	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
phaseDeviation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
progress	INTEGER	R/O	IntegerRange R/O (0 .. 100)
longitude	TEXT	R/O	StringRange R/O
latitude	TEXT	R/O	StringRange R/O
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lsats	TEXT	R/O	StringRange R/O
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (GNSS receive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Antenna connection)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Antenna shortcut)

9.687 TimeLink-GPS-Receiver

This page describes the monitoring only device driver and the device window for the TimeLink GPS Frequency and Time Generator.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The GPS related parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field

is mandatory.

- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
longitude	TEXT	R/O	StringRange R/O
latitude	TEXT	R/O	StringRange R/O
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
utcTime	TEXT	R/O	StringRange R/O
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (GPS receive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Antenna over current)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Antenna under current)

9.688 TimeLink-TMG3420

This page describes the monitoring only device driver and the device window for the TimeLink GPS Frequency and Time Generator TMG3420 via SNMP.

It was tested with model: TMG3420

- LO: LO0792A1
- FPGA XL00776A1
- MIB 1.0.4

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The GNSS related parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	**description**
address	The SNMP address specification for the device (details below).
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	**description**

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.identifier	TEXT	R/O	StringRange R/O
info.mibVersion	TEXT	R/O	StringRange R/O
info.aging	TEXT	R/O	StringRange R/O
info.equipmentLO	TEXT	R/O	StringRange R/O
info.fpga	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.uniqueId	TEXT	R/O	StringRange R/O
info.securityLevel	TEXT	R/O	StringRange R/O
info.board	TEXT	R/O	StringRange R/O
info.oscillator	TEXT	R/O	StringRange R/O
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.alarmCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.alarmList	TEXT	R/O	StringRange R/O
utcTime	TEXT	R/O	StringRange R/O
state.frequency	TEXT	R/O	StringRange R/O
state.time	TEXT	R/O	StringRange R/O
state.stability	TEXT	R/O	StringRange R/O
oscillator.dac	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
phaseDeviation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
progress	INTEGER	R/O	IntegerRange R/O (0 .. 100)
longitude	TEXT	R/O	StringRange R/O
latitude	TEXT	R/O	StringRange R/O
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
nsats	INTEGER	R/O	IntegerRange R/O (0 .. 0)
lsats	TEXT	R/O	StringRange R/O
rxok	BOOLEAN	R/O	BooleanRange R/O (true false)
state.timesource	TEXT	R/O	StringRange R/O
state.phasing	TEXT	R/O	StringRange R/O
state.tracking	TEXT	R/O	StringRange R/O
state.tfom	TEXT	R/O	StringRange R/O
state.ete	TEXT	R/O	StringRange R/O
state.oscillator	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (IRIGB no signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (IRIGB signal decode)
faults.03	ALARM	R/O	AlarmFlagRange R/O (IRIGB signal wrong)
faults.04	ALARM	R/O	AlarmFlagRange R/O (IRIGB Decoder)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Free running expired)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Files signatures)
faults.07	ALARM	R/O	AlarmFlagRange R/O (UTC offset)
faults.08	ALARM	R/O	AlarmFlagRange R/O (GNSS antenna n.c.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (GNSS antenna S/C)
faults.10	ALARM	R/O	AlarmFlagRange R/O (GPS No satellites)
faults.11	ALARM	R/O	AlarmFlagRange R/O (GPS Test mode)
faults.12	ALARM	R/O	AlarmFlagRange R/O (GPS Position)
faults.13	ALARM	R/O	AlarmFlagRange R/O (GPS no stored pos.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Spoofing shift)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Spoofing discontinuity)
faults.16	ALARM	R/O	AlarmFlagRange R/O (GNSS time)
faults.17	ALARM	R/O	AlarmFlagRange R/O (No NMEA signal)
faults.18	ALARM	R/O	AlarmFlagRange R/O (NMEA checksum)
faults.19	ALARM	R/O	AlarmFlagRange R/O (NMEA out of range)

name	*type*	*flags*	*range*
faults.20	ALARM	R/O	AlarmFlagRange R/O (Power 1)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Power 2)
faults.22	ALARM	R/O	AlarmFlagRange R/O (OCXO 10MHz)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Output 1 S/C)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Output 2 S/C)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Output 3 S/C)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Output 4 S/C)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Output 5 S/C)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Output 6 S/C)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Output 7 S/C)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Output 8 S/C)
faults.31	ALARM	R/O	AlarmFlagRange R/O (GNSS survey running)
faults.32	ALARM	R/O	AlarmFlagRange R/O (GNSS leap sec pending)

9.689 Timetech-Distributor

This page describes the device driver and the device window for Timetech Frequency Distributor

The Unit sends Module Status Message (starting with %DM) and Unit Status Message (%0) every 10 seconds. Therefore one GET procedure catches all inputs.

The driver was tested with the following model:

- TIM-10116

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The switch parameters
- --- The faults page.
- --- The device info page.

- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.

Remote Control

TCP Port 2000 Please set the TCP connection timeout to > 10sec (successfully tested with 15sec) to avoid read timeouts. **Attention:**

The network interface seems to be unstable under rare circumstances and the unit may crash. Recommendation: Use the serial interface instead. **Unit Configuration**

Please set ECHO=0 within units saterm.ini file. This disables the echo of commands.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
protection	CHOICE		EnumRange (OFF ON ONCE)
location	CHOICE		EnumRange (1 2)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
internal.msg	TEXT	R/O	StringRange R/O
info.module	TEXT	R/O	StringRange R/O
info.unit	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Module 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Module 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Module 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module 4)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Module 5)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Module 6)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Module 7)

9.690 Timetech-GPS-Receiver

This page describes the device driver and the device window for Timetech Reference Generator

The Unit sends Module Status Message (starting with %DM) and Unit Status Message (%0) every 10 seconds. Therefore one GET procudure catches all inputs.

The driver was tested with the following model:

- TIM-10292

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- GPS position data
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
freqSource	Defines the used frequency source.

Remote Control

TCP Port 2000 Please set the TCP connection timeout to > 10sec (successfully tested with 15sec) to avoid read timeouts. **Unit configuration**

Please set ECHO=0 in units saterm.ini. This disables the echo of commands.**Attention**

The network interface seems to be unstable under rare circumstances and the unit may crash. Recommendation: Use the serial interface instead.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
internal.msg	TEXT	R/O	StringRange R/O
info.module	TEXT	R/O	StringRange R/O
info.unit	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.freqSource	CHOICE	SETUP	EnumRange (GPS PPS-1 PPS-2 OPLINK PPSPC EXT10 PULSE)
longitude	TEXT	R/O	StringRange R/O
latitude	TEXT	R/O	StringRange R/O
altitude	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state	TEXT	R/O	StringRange R/O
timeMode	CHOICE	R/O	EnumRange R/O (GPS UTC)
source	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (GPS receive)
faults.02	ALARM	R/O	AlarmFlagRange R/O (NTP receive)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Module 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Module 2)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Module 3)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Module 4)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Module 5)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Module 6)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Module 7)

9.691 Toggle-Switch

This page describes the device driver and the device window for the*Toggle-Switch* logical device. This device controls a latching relay providing only one drive coil and one or two position indication circuits. Such a switch works like a ball pen mechanics, each current pulse to the drive coil toggles the switch position once.

The *Toggle-Switch* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

The device supports the usage of one or two position indication ports which tell the device in

which position the switch actually is. You may define either the indicationAPort or the indicationBPort, depending on the polarity of the signal. If there are both position indication inputs configured, the device checks these to be oppositional, states an position indication fault if they are equal.

The *Toggle-Switch* logical device does not check if the commanded position matches the indicated position. You may use the device to generate an output pulse without reading back anything. For this, configure one of the indication ports to an arbitrary input signal and ignore the 'position' value. Instead use the 'toggle' parameter which generates an output pulse any time it gets written.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets / shows the actual switch state.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- drivePort --- The address of the PLC port which toggles the switch position. See the paragraph 'Port addressing' below for details.
- indicationAPort --- The address of the 'position A' indication port, which tells that the relay is in "A" position. May be left empty, if indicationBPort is defined.
- indicationBPort --- The address of the 'position B' indication port, which tells that the relay is in "B" position. May be left empty if indicationAPort is defined.
- pulseWidth --- The activation pulse width (msecs) to be sent to the switch for a position change. When setting this parameter, you should be aware that pulses which are shorter than the IDLE time of the PLC driver's thread may get lost. If in doubt, set the pulse width to 1100 msecs or in the VLC interface/device setup screen tune the IDLE time of the PLC driver thread to e.g. 200 msecs.
- positions --- By default the device reports the state of the output as 'A'/'B'. You may specify arbitrary texts for these states. 'positions' contains the two descriptions, separated by a comma. Please note, that the graphical "Switch" element at the sat-nms user interface *requires* the switch to use the 'A'/'B' selections.
- enableQuery --- Selects if the switch position selector at the first page of the device window shall query the operator before it actually switches. Default value is "NO".Please note, that changing this parameter only applies to device windows which are opened *after* the parameter has been changed.

- commFaultId --- By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault. The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).

Fault flags

fault flag	***fault name***	***description***
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Position Indication	This fault is set if the switch position cannot be determined unambiguously. This is if not exactly one of the indication ports is 'high'.
faults.02	Configuration Setup	This fault signals that the at least one of the xxxxPort parameters listed above ist not properly configured. The device requires that both drive ports and both position indication ports are set to values which matches the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the **+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address ' *FANUC.Q00065+1* ' refers to the *2^1* bit in the variable ' *Q00065* ' provided by the ' *FANUC* ' device. In fact this addresses the output table register '%Q00066' at the PLC. Other PLC drivers may use a different naming scheme for the variables, but the

concept always is the same.

Switch positions

The *Toggle-Switch* logical device internally controls and monitors the switch position as A/B, following the names of the port configuration parameters *indicationAPort* and *indicationBPort* . This internal (not translated) switch position may differ from the switch position signalled with the *position* selection at the device.

Using the *positions* configuration parameter, the switch positions may be renamed to arbitrary values. E.g. the positions of a switch may be labelled "DUMMY LOAD" and "ANTENNA". You should however consider, that the graphical "Switch" element at the sat-nms user interface only works, if the selections for the switch are names "A" and "B".

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.indication	TEXT	R/O	StringRange R/O
config.drivePort	TEXT	SAVE SETUP	StringRange
config.indicationAPort	TEXT	SAVE SETUP	StringRange
config.indicationBPort	TEXT	SAVE SETUP	StringRange
config.pulseWidth	INTEGER	SAVE SETUP	IntegerRange (200 .. 5000)
config.positions	TEXT	SAVE SETUP	StringRange
config.enableQuery	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (A B)

name	*type*	*flags*	*range*
toggle	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Position Indication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.692 Transmit-Chain-Preset

The Transmit-Chain-Preset logical device manages to store / retrieve the settings of a complete transmit chain in one preset. The device collects the setting of a polarization switch, the frequency and the EIRP of the transmission and the device presets of two arbitrary devices (e.g. modulator and encoder) into one DevicePreset.

When such a preset is applied to the Transmit-Chain-Preset logical device, it forwards the stored settings to the appropriate devices.

The Transmit-Chain-Preset device stores it's presets always in the same directory. In an M&C configuration this usually is what you expect. There probably is one transmit chain, or two almost identical ones, all presets stored from there apply to the same device configuration. When used in an NMS environment, the transmit chain presets of all VLCs are stored in the same directory. If the VLCs are configured for different device setups, you should name the presets in a way that presets for different device setups are not mixed up. Applying a transmit chain preset to a VLC with a different device setup as it was recorded, will cause incomplete or wrong device settings.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameter which must be set to interconnect the Transmit-Chain-Preset device with the devices it shall collect the settings from.

parameter	*description*

parameter	*description*
switchId	Enter the message identifier here of the polarization setting the device shall store in the preset.
frequencyId	Enter the message identifier here of the frequency value the device shall store in the preset.
eirpId	Enter the message identifier here of the EIRP or gain setting the device shall store in the preset.
device1Name	The name of the first device which shall store all it's settings into the preset
device2Name	The name of the second device which shall store all it's settings into the preset

Remarks

With version 2.00 of the driver the Transmit-Chain-Preset device has been adapted for device presets stored in text format. Unlike other devices, the Transmit-Chain-Preset allots the setting by the order of appearance in the file. Hence never add or remove lines of a Transmit-Chain-Preset file, even empty lines or comments will make the file unreadable.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.switchId	TEXT	SAVE SETUP	StringRange
config.frequencyId	TEXT	SAVE SETUP	StringRange
config.eirpId	TEXT	SAVE SETUP	StringRange
config.device1Name	TEXT	SAVE SETUP	StringRange
config.device2Name	TEXT	SAVE SETUP	StringRange

9.693 Triple-Translator

The Triple-Translator logical device translates a set of user definable choices into the position

settings of three switches (or equivalent). It may be used to combine three switches / parameter settings to a single function in the user interface.

To make use of the Triple-Translator, install it in the device setup, configure the translation tables and the message ID you want to translate. In your user screens operate the translated parameter now through the 'value' parameter of the Triple-Translator instead of directly via the parameters original message ID.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Simply contains a choice box to operate the translated parameter. Provided for completeness and for testing purposes.
- --- The device info page.
- --- The maintenance page. The configuration parameters from this page are described below.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Triple-Translator device with the device / parameter it shall translate.

parameter	***description***
choices	Enter the list of choices here which are to be presented to the operator when he edits the 'value' setting of this Triple-Translator device. The choice list must be entered as a comma separated list of tokens with no additional spaces between them.
messageId1	Enter the message identifier here of the first parameter / switch setting which shall be controlled, eg. "SWITCH-1.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
choices1	Enter the list of choices/positions here for the first switch. There must be the same number of position settings contained in the list as choices defined above. Choices and switch positions are assigned by the order they appear in their lists.
messageId2	Enter the message identifier here of the second parameter / switch setting which shall be controlled, eg. "SWITCH-2.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.

parameter	*description*
choices2	Enter the list of choices/positions here for the second switch. There must be the same number of position settings contained in the list as choices defined above. Choices and switch positions are assigned by the order they appear in their lists.
messageId3	Enter the message identifier here of the third parameter / switch setting which shall be controlled, eg. "SWITCH-3.position". Look at the list of device driver variable definitions to see correct name of this setting at the device type you are using.
choices3	Enter the list of choices/positions here for the third switch. There must be the same number of position settings contained in the list as choices defined above. Choices and switch positions are assigned by the order they appear in their lists.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.choices	TEXT	SAVE SETUP	StringRange
config.messageId1	TEXT	SAVE SETUP	StringRange
config.choices1	TEXT	SAVE SETUP	StringRange
config.messageId2	TEXT	SAVE SETUP	StringRange
config.choices2	TEXT	SAVE SETUP	StringRange
config.messageId3	TEXT	SAVE SETUP	StringRange
config.choices3	TEXT	SAVE SETUP	StringRange
value	CHOICE		EnumRange ()

9.694 TrippLite-PDU

This page describes the device driver and the device window for the Tripp-Lite power distribution unit including 16 outlets

The driver was tested with the following model:

- Model: PDUMH32HVNET
- SW: 6925750A

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all outlets names and state.
- --- Shows all advance outlet measurements incl. reset, ramp and shed commands.
- --- Shows all in and output measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device. The default community string is *tripplite*
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. The default community string is *tripplite* . If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.inputs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.phase	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.circuit	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.breakers	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.heatsinks	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.outlets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
output.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
output.current.min	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.current.max	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
ramp	TEXT	NOPRESET	StringRange
shed	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
input.voltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.frequency	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
input.switchpoint	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.loVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
input.hiVoltage	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
circuit.current	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.01	CHOICE		EnumRange (NONE ON OFF)
state.02	CHOICE		EnumRange (NONE ON OFF)
state.03	CHOICE		EnumRange (NONE ON OFF)
state.04	CHOICE		EnumRange (NONE ON OFF)
state.05	CHOICE		EnumRange (NONE ON OFF)
state.06	CHOICE		EnumRange (NONE ON OFF)
state.07	CHOICE		EnumRange (NONE ON OFF)
state.08	CHOICE		EnumRange (NONE ON OFF)
state.09	CHOICE		EnumRange (NONE ON OFF)
state.10	CHOICE		EnumRange (NONE ON OFF)
state.11	CHOICE		EnumRange (NONE ON OFF)
state.12	CHOICE		EnumRange (NONE ON OFF)
state.13	CHOICE		EnumRange (NONE ON OFF)
state.14	CHOICE		EnumRange (NONE ON OFF)
state.15	CHOICE		EnumRange (NONE ON OFF)
state.16	CHOICE		EnumRange (NONE ON OFF)
state.17	CHOICE		EnumRange (NONE ON OFF)
state.18	CHOICE		EnumRange (NONE ON OFF)
state.19	CHOICE		EnumRange (NONE ON OFF)
state.20	CHOICE		EnumRange (NONE ON OFF)
state.21	CHOICE		EnumRange (NONE ON OFF)
state.22	CHOICE		EnumRange (NONE ON OFF)

name	*type*	*flags*	*range*
state.23	CHOICE		EnumRange (NONE ON OFF)
state.24	CHOICE		EnumRange (NONE ON OFF)
name.01	TEXT		StringRange
name.02	TEXT		StringRange
name.03	TEXT		StringRange
name.04	TEXT		StringRange
name.05	TEXT		StringRange
name.06	TEXT		StringRange
name.07	TEXT		StringRange
name.08	TEXT		StringRange
name.09	TEXT		StringRange
name.10	TEXT		StringRange
name.11	TEXT		StringRange
name.12	TEXT		StringRange
name.13	TEXT		StringRange
name.14	TEXT		StringRange
name.15	TEXT		StringRange
name.16	TEXT		StringRange
name.17	TEXT		StringRange
name.18	TEXT		StringRange
name.19	TEXT		StringRange
name.20	TEXT		StringRange
name.21	TEXT		StringRange
name.22	TEXT		StringRange
name.23	TEXT		StringRange
name.24	TEXT		StringRange

name	*type*	*flags*	*range*
ramp.01	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.02	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.03	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.04	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.05	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.06	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.07	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.08	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.09	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.10	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.11	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.12	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.13	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.14	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.15	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.16	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.17	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.18	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.19	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.20	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.21	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.22	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.23	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.24	CHOICE		EnumRange (REMAIN OFF TURN ON)
ramp.01.delay	INTEGER		IntegerRange (0 .. 0)
ramp.02.delay	INTEGER		IntegerRange (0 .. 0)
ramp.03.delay	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ramp.04.delay	INTEGER		IntegerRange (0 .. 0)
ramp.05.delay	INTEGER		IntegerRange (0 .. 0)
ramp.06.delay	INTEGER		IntegerRange (0 .. 0)
ramp.07.delay	INTEGER		IntegerRange (0 .. 0)
ramp.08.delay	INTEGER		IntegerRange (0 .. 0)
ramp.09.delay	INTEGER		IntegerRange (0 .. 0)
ramp.10.delay	INTEGER		IntegerRange (0 .. 0)
ramp.11.delay	INTEGER		IntegerRange (0 .. 0)
ramp.12.delay	INTEGER		IntegerRange (0 .. 0)
ramp.13.delay	INTEGER		IntegerRange (0 .. 0)
ramp.14.delay	INTEGER		IntegerRange (0 .. 0)
ramp.15.delay	INTEGER		IntegerRange (0 .. 0)
ramp.16.delay	INTEGER		IntegerRange (0 .. 0)
ramp.17.delay	INTEGER		IntegerRange (0 .. 0)
ramp.18.delay	INTEGER		IntegerRange (0 .. 0)
ramp.19.delay	INTEGER		IntegerRange (0 .. 0)
ramp.20.delay	INTEGER		IntegerRange (0 .. 0)
ramp.21.delay	INTEGER		IntegerRange (0 .. 0)
ramp.22.delay	INTEGER		IntegerRange (0 .. 0)
ramp.23.delay	INTEGER		IntegerRange (0 .. 0)
ramp.24.delay	INTEGER		IntegerRange (0 .. 0)
shed.01	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.02	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.03	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.04	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.05	CHOICE		EnumRange (REMAIN ON TURN OFF)

name	*type*	*flags*	*range*
shed.06	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.07	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.08	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.09	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.10	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.11	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.12	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.13	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.14	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.15	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.16	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.17	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.18	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.19	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.20	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.21	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.22	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.23	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.24	CHOICE		EnumRange (REMAIN ON TURN OFF)
shed.01.delay	INTEGER		IntegerRange (0 .. 0)
shed.02.delay	INTEGER		IntegerRange (0 .. 0)
shed.03.delay	INTEGER		IntegerRange (0 .. 0)
shed.04.delay	INTEGER		IntegerRange (0 .. 0)
shed.05.delay	INTEGER		IntegerRange (0 .. 0)
shed.06.delay	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
shed.07.delay	INTEGER		IntegerRange (0 .. 0)
shed.08.delay	INTEGER		IntegerRange (0 .. 0)
shed.09.delay	INTEGER		IntegerRange (0 .. 0)
shed.10.delay	INTEGER		IntegerRange (0 .. 0)
shed.11.delay	INTEGER		IntegerRange (0 .. 0)
shed.12.delay	INTEGER		IntegerRange (0 .. 0)
shed.13.delay	INTEGER		IntegerRange (0 .. 0)
shed.14.delay	INTEGER		IntegerRange (0 .. 0)
shed.15.delay	INTEGER		IntegerRange (0 .. 0)
shed.16.delay	INTEGER		IntegerRange (0 .. 0)
shed.17.delay	INTEGER		IntegerRange (0 .. 0)
shed.18.delay	INTEGER		IntegerRange (0 .. 0)
shed.19.delay	INTEGER		IntegerRange (0 .. 0)
shed.20.delay	INTEGER		IntegerRange (0 .. 0)
shed.21.delay	INTEGER		IntegerRange (0 .. 0)
shed.22.delay	INTEGER		IntegerRange (0 .. 0)
shed.23.delay	INTEGER		IntegerRange (0 .. 0)
shed.24.delay	INTEGER		IntegerRange (0 .. 0)
pulse.01	TEXT	NOPRESET	StringRange
pulse.02	TEXT	NOPRESET	StringRange
pulse.03	TEXT	NOPRESET	StringRange
pulse.04	TEXT	NOPRESET	StringRange
pulse.05	TEXT	NOPRESET	StringRange
pulse.06	TEXT	NOPRESET	StringRange
pulse.07	TEXT	NOPRESET	StringRange
pulse.08	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
pulse.09	TEXT	NOPRESET	StringRange
pulse.10	TEXT	NOPRESET	StringRange
pulse.11	TEXT	NOPRESET	StringRange
pulse.12	TEXT	NOPRESET	StringRange
pulse.13	TEXT	NOPRESET	StringRange
pulse.14	TEXT	NOPRESET	StringRange
pulse.15	TEXT	NOPRESET	StringRange
pulse.16	TEXT	NOPRESET	StringRange
pulse.17	TEXT	NOPRESET	StringRange
pulse.18	TEXT	NOPRESET	StringRange
pulse.19	TEXT	NOPRESET	StringRange
pulse.20	TEXT	NOPRESET	StringRange
pulse.21	TEXT	NOPRESET	StringRange
pulse.22	TEXT	NOPRESET	StringRange
pulse.23	TEXT	NOPRESET	StringRange
pulse.24	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Load 1 Off)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Load 2 Off)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Load 3 Off)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Load 4 Off)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Load 5 Off)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Load 6 Off)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Load 7 Off)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Load 8 Off)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Load 9 Off)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Load 10 Off)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (Load 11 Off)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Load 12 Off)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Load 13 Off)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Load 14 Off)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Load 15 Off)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Load 16 Off)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Load 17 Off)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Load 18 Off)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Load 19 Off)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Load 20 Off)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Load 21 Off)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Load 22 Off)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Load 23 Off)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Load 24 Off)

9.695 Uplink-Power-Control-2

The Uplink-Power-Control-2 logical device adjusts the level of a transmit signal in order to compensate the atmospheric attenuation in the signal uplink. To achieve this, the Uplink-Power-Control logical device reads the level of the satellite's beacon from a beacon receiver or from another signal source. Unlike the Uplink-Power-Control device, which is optimized for a permanent long term transmission, the Uplink-Power-Control-2 device is intended to be used with occasional use transmissions.

Functional description

The device controls and adjusts a transmit level by adding the deviation between a measured beacon level or signal value read from another uplink power control and its nominal value to a given nominal transmit power. A new power value is set every time either the measured level or one of the operational parameters change.

Operational parameters

There are a number of operational parameters which can be used to tune the behaviour of the

Uplink-Power-Control-2 device:

parameter	***description***
nominalOutput	The nominal transmit power value (usually in units of EIRP). This value must be set to make the uplink power control work.
nominalInput	The nominal value of the receive level or what reference value is used for the uplink power control.
upcEnable	Switches the uplink power control on or off. Switching OFF sets the output power to its nominal value.
maxPwrIncrease	Limits the increase of the output power with respect to the nominal value (dB).
maxPwrDecrease	Limits the decrease of the output power with respect to the nominal value (dB). If the nominal values refer to clear sky conditions, this value wil be 0 in most cases (no decrease of the output power).
ratio	The deviation of the measured input value is multiplied with this ratio setting before the result is added to the nominal output power. Hence, the ratio value fan be used to control the direction and the intensity of the uplink power control (see section 'Control ratio' below).

Power on behaviour

The Uplink-Power-Control-2 device does not permanently store the nominal transmit power level. When the sat-nms server starts, the nominal transmit power is unknown and the uplink power control is inhibited until this value is set. This can be done by the operator by setting this value explicitly or by another logical device which commands a transmit power for a new job.

Control ratio

The 'ratio' parameter is used to adapt the uplink power control behaviour to the kind of reference value used as input.

If the input is a beacon level, the ratio value must be negative (e.g. -1.0). If the beacon level falls below the nominal value, the output power shall increase.

The uplink power control also my be operated as a slave to another power control. If the deviation of the output power of this other uplink power control device is used, the ration parameter must be positive (+1.0). If the other uplink power control raises the output power, we do as well.

When used with a beacon level, the device has to estimate the atmospheric attenuation in the uplink from a lowering of the beacon (downlink) level. As transmission usually is done at higher frequencies than the downlink, the uplink atmospheric attenuation will be higher than the measured beacon level degradation. This might be a reason to set the ratio value to a value

like -1.2 in order to cover this effect.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational parameters.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
pwrSetId	The parameter ID which controls the transmit level. In most cases this will be the 'eirp' parameter of the Level-Set or Level-Control logical device which controls the transmit power. If no Level-Set or Level-Control device is used, a modulator level or upconverter gain parameter may be used as well.
pwrMeasId	The ID of the parameter indicating the satellite beacon level (or another reference value to be used).
enableId	The ID of a parameter which enables the UPC. If this parameter is configured, the UPC works only if upcEnable is set ON, *and* the parameter addressed by this enableId is one of 'true', 'YES' or 'ON'.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.pwrSetId	TEXT	SAVE SETUP	StringRange
config.pwrMeasId	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.enableId	TEXT	SAVE SETUP	StringRange
nominalOutput	FLOAT		DoubleRange (0.00 .. 0.00)
nominalInput	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
upcEnable	CHOICE	SAVE	EnumRange (OFF ON)
maxPwrIncrease	FLOAT	SAVE	DoubleRange (0.00 .. 60.00)
maxPwrDecrease	FLOAT	SAVE	DoubleRange (0.00 .. 60.00)
ratio	FLOAT	SAVE	DoubleRange (-5.00 .. 5.00)
inputDelta	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
outputDelta	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inputPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
outputPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
enabled	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power decrease limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power increase limit)

9.696 Uplink-Power-Control

The Uplink-Power-Control logical device adjusts the level of a transmit signal in order to compensate the atmospheric attenuation in the signal uplink. To achieve this, the Uplink-Power-Control logical device reads the level of the satellite's beacon from a beacon receiver. The decrease of the beacon level with respect to it's 'clear sky' value describes the degree of downlink atmospheric attenuation which is used to estimate the atmospheric attenuation in the uplink.

The Uplink-Power-Control logical device must not use the level indication of the signal received back from the satellite instead of the beacon level. This would create a 'closed loop' level control which tends to oscillate or at least overshoot in an impermissible way.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the operational parameters.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to interconnect the Level-Set device to the devices it monitors and controls.

parameter	***description***
pwrSetId	The parameter ID which controls the transmit level. In most cases this will be the 'eirp' parameter of the Level-Set or Level-Control logical device which controls the transmit power. If no Level-Set or Level-Control device is used, a modulator level or upconverter gain parameter may be used as well.
pwrMeasId	The ID of the parameter indicating the satellite beacon level.
txFrequencyId	You may enter a parameter ID here which reports the transmit frequency if you want make the device calculate the control ratio automatically. Please note that the frequency parameter must report the true RF transmit frequency in MHz. IF or L-band frequency settings cannot be used for this.
rxFrequencyId	You may enter a parameter ID here which reports the satellite beacon frequency if you want make the device calculate the control ratio automatically. Please note that the frequency parameter must report the true RF frequency of the beacon in MHz. IF or L-band frequency settings cannot be used for this.
resetOnDisable	This parameter controls the behaviour of the UPC when it gets disabled. When resetOnDisable is set to 'NO' (this is the default setting), then the output power will stay at it's last value before the UPC was switched OFF. With resetOnDisable set to 'YES' the output power will be reset to it's nominal value if the UPC ist switched off.
inhibitId	If this parameter is not empty, the device listens to the parameter with this ID and inhibits the control loop if the parameter value matches the value of 'inhibitVal'
inhibitVal	If an 'inhibitId' is given, this parameter should contain the value which triggers the inhibit state. Without an 'inhibitId' this parameter is ignored.

parameter	*description*
inhibitMode	May be one of 'LATCHING' or 'MOMENTARY'. With LATCHING (which is the default) the device remains inhibited even if the inhibit condition is no longer present. Only a reset command clears the inhibit state in this mode. With MOMENTARY the inhibit data gets automatically cleared if the inhibit condition disappears.
ltFaultDelay	This parameter defines the time (minutes) the 'power increase' fault must be constantly active to set the 'long term power increase' fault
logLevelEnableId	This parameter can be used to control the logging of gain/level changes at the controlled HPA in a way, that the logging gets suppressed while the UPC is running. This prevents the log from being flooded with parameter change messages caused by the level regulation of the UPC. When the UPC is disabled, the logging is enabled again. If this parameter is not empty, the device send the value "OFF" to this ID when the upcEnable is set to "ON" and vice versa.
redundancyOffset	This offset gets added to the nominal output level of this device and then set as nominal output of the redundant UPC by a Protection-Switch-N-To-1. Set this to the level difference the redundant path has with respect to this path.

Clear Sky Settings

Before the uplink power control can work, it must know the nominal values for the transmit power and the beacon level under clear sky conditions. You may enter these values to the appropriate fields at the primary device window page of the device. However, the more comfortable way is to press the 'CLEAR SKY' button once the transmit level has been setup properly. The this function also may be used to re-adjust the nominal values from time to time in order to compensate for some gain drift.

Control Ratio

The device has to estimate the atmospheric attenuation in the uplink from a lowering of the beacon (downlink) level. As transmission usually is done at higher frequencies than the downlink, the uplink atmospheric attenuation will be higher than the measured beacon level degradation.

The 'ratio' parameter found at the 'Advanced Parameters' page defines the uplink/downlink atmospheric attenuation ratio to be applied. You may set this value manually or let the device calculate it from the transmit / receive frequencies (ratio mode = 'AUTO'). E.g. for the Ku-band a ratio of 1.2 may be used which means that the transmit power gets increased by 1.2 dB if the beacon level falls by 1 dB.

When using the automatic mode, the device calculates the ratio from the frequency values shown at the same page. If you did set the 'txFrequencyId' and 'rxFrequencyId' configuration

parameters, the frequency value are read automatically from the real settings. If you didn't configure this, you have to enter the frequency values manually.

Filtering And Hysteresis

The device provides some parameters to adjust the behaviour of the control that it does not react on every glitch in the beacon level. These parameter are:

- **clear sky hysteresis:** The control will only increase the transmit level if the beacon level is at least this value below it's nominal reading. A typical value for this setting is 0.3 dB. The uplink power control ignores the natural jitter of the beacon level and starts to work if the beacon level falls to 0.3 dB below the clear sky value.
- **output hysteresis:** Once the uplink power control starts to work, it adjusts the transmit level with every tenth of a dB the beacon level changes. If you want that the transmit level is only adjusted when the value changes significantly, set this hysteresis parameter to a non-zero value.
- **filter:** The device also includes a low pass filter which may be applied to the beacon level readings. Please note, that activating the filter always will introduce an additional delay to the processing in the device.

Output Power Slope Limit

The device is capable to limit the slope of the commanded output power to user defined values. This means, a leap of the input power does not lead to an output power leap, the output power is commanded step by step to reach its target value with the given slope.

Power increment / decrement slopes are defined in dB/sec, the device internally interprets this as dB / driver cycle. Please ensure, that the IDLE time of the interface containing the Uplink-Power-Control device is set to 1000 msecs to match 'dB/sec' and 'dB / driver cycle'.

To enable the slope limits, set the parameter 'speed limit' to 'YES' and the slope values as desired.

Limits And Fault Detection

The device provides two parameters to limit the amount of transmit power adjustment:

- **max power increase:** The value defines the maximum increase the uplink power control applies to the transmit level. If the beacon level falls in a way that would lead to a higher transmit power, the latter is commanded to the highest possible value and a 'High limit' fault is raised. The fault is cleared as soon as the limit condition disappears.
- **max power decrease:** This limit value works the same way, but for transmit level *decreases* . Usually you will set this to zero in order to prevent the device from decreasing the transmit power below the nominal value.

Additionally, the device reports a 'long term power increase' fault which becomes active, when the 'power increase fault' is constantly active for a given time. This fault is latching am mut be reset by the 'RST FLT' button to be cleared. The duration how long the 'power increase fault' must be active to trigger this fault is tunable as a configuration parameter.

Beside the limit values discussed above, the device provides a parameter called**drop threshold** . If this parameter is set to a non zero value, the device permanently monitors the change of the beacon level from one reading to the next. If the beacon level suddenly falls by an amount larger than the threshold set, the uplink power control gets stopped *before* this dubitable value is processed. The device raises a 'Drop detected' fault and stops working until the RESET button ist pressed to clear the fault. This behaviour shall prevent the device from setting the transmit power to it's maximum in case of a failure or maloperation of the beacon receiver.

Inhibiting the UPC

The device is capable to watch an external parameter value and inhibit the UPC processing if this parameter turns to a certain value. This way for example a LNB fault can be watched and used to to inhibit the control loop in case of an error.

The inhibit state is latching, it must be cleared manually by pressing 'RST INH'. Depending on the 'resetOnDisable' mode, the control loop freezes when inhibited (resetOnDisable=NO) or reverts to the nominal output power (resetOnDisable=YES) when inhibited.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.pwrSetId	TEXT	SAVE SETUP	StringRange
config.pwrMeasId	TEXT	SAVE SETUP	StringRange
config.txFrequencyId	TEXT	SAVE SETUP	StringRange
config.rxFrequencyId	TEXT	SAVE SETUP	StringRange
config.resetOnDisable	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.inhibitId	TEXT	SAVE SETUP	StringRange
config.inhibitVal	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inhibitMode	CHOICE	SAVE SETUP	EnumRange (LATCHING MOMENTARY)
config.ltFaultDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.logLevelEnableId	TEXT	SAVE SETUP	StringRange
config.redundancyOffset	FLOAT	SAVE SETUP	DoubleRange (-10.00 .. 10.00)
nominalOutput	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
nominalInput	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
upcEnable	CHOICE	SAVE	EnumRange (OFF ON)
maxPwrIncrease	FLOAT	SAVE	DoubleRange (0.00 .. 60.00)
maxPwrDecrease	FLOAT	SAVE	DoubleRange (0.00 .. 60.00)
maxPwrIncSpeed	FLOAT	SAVE	DoubleRange (0.1 .. 10.0)
maxPwrDecSpeed	FLOAT	SAVE	DoubleRange (0.1 .. 10.0)
clearSkyHysteresis	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
outputHysteresis	FLOAT	SAVE	DoubleRange (0.00 .. 10.00)
filter	CHOICE	SAVE	EnumRange (OFF 1 Hz 0.5 Hz 0.2 Hz 0.1 Hz 0.05 Hz 0.02 Hz 0.01 Hz 0.005 Hz 0.002 Hz 0.001 Hz)
ratio	FLOAT	SAVE	DoubleRange (0.00 .. 5.00)
ratioMode	CHOICE	SAVE	EnumRange (MANUAL AUTO)
limitPwrSpeed	CHOICE	SAVE	EnumRange (NO YES)
txFrequency	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
rxFrequency	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
dropThreshold	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
clearSkyNow	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
clearInhibit	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
clearLtFault	TEXT	NOPRESET	StringRange
inputDelta	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
outputDelta	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
inputPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
outputPower	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
adjusting	BOOLEAN	R/O	BooleanRange R/O (true false)
inhibited	BOOLEAN	R/O	BooleanRange R/O (true false)
saved.faults.03	BOOLEAN	SAVE	BooleanRange (true false)
saved.faults.06	BOOLEAN	SAVE	BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power decrease limit)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power increase limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Drop detected)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Long term pwr increase)
faults.05	ALARM	R/O	AlarmFlagRange R/O (UPC Off)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Inhibited)

9.697 V4L-DVBS2-Demod

This page describes the device driver and the device window for V4L (video for linux) based DVB-S2 receiver cards. The driver has been tested with a TurboSight TBS6981 dual receiver card, it may work also with other cards based on V4L but this cannot be guaranteed.

The sat-nms software operates this card as a receive quality measurement device only. No demuxer, or video/audio decoder is included with the software used for this card.

ATTENTION

TurboSight TBS6981 cards require a 64bit Linux system and a special kernel module to be installed. The cards will not work on a "standard" VLC/M&C system.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the receive settings like frequency and symbol rate.
- --- Shows the actual demodulator state including the figures of receive level and quality.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to this device.

parameter	***description***
address	Leave this field empty, the protocol address is not used by the device driver. Each individual demodulator unit is addressed by the interface address specified in the device configuration screen (See 'installation' at the end of this page).
22kHz	Makes the device apply a 22 kHz tone to the LNB power when set to ON.
22kHzAuto	If set to ON, the driver switches the 22 kHz tone depending on the receive frequency. Frequencies above the 'bandEdge' value cause the 22kHz tone to be switched on, lower frequencies switch the tone off. The 22kHzAuto mode overwrites the the '22kHz' setting described above.
lnbPower	Selects if the device shall apply LNB power to the input.
lnbVoltage	Selects the LNB power supply voltage (13V/18V). This switches the polarization (13V=Vertical) with some LNBs.
lnbPolSwitch	If you set this parameter to ON, the 'lnbVoltage' parameter above becomes read-only and the LNB voltage is controlled though the 'polarization' parameter of the device driver (only makes sense with lnbPower=ON).
loFreqHigh	The LO frequency for the high frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.

parameter	*description*
loFreqLow	The LO frequency for the low frequency band. The software sets the appropriate LO frequency every time before it changes the receive frequency.
bandEdge	The frequency where the software switches the LO frequencies.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Installation

The TBS6981 is a dual DVB-S2 receiver, hence there are two instances of the driver to be configured for each card.

When configuring a V4L-DVBS2-Demod device in the device configuration screen, you have to specify the card's adapter/frontend number in the linux /dev/dvb directory tree instead of the device's interface name. The address consists of 2 digits *"x.y"*, where:

- *x* is the adapter number, starting a 0
- *y* is the frontend number, starting a 0

Example: *"1.0"* specifies the first frontend on the second card in the computer.

A TBS6981 card appears as two adapters with one frontend each, thus giving the addresses *"0.0"*, *"1.0"* for the first card and *"2.0"*, *"3.0"* for the second card in the system.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serialno	TEXT	R/O	StringRange R/O
info.cardid	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.22kHz	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.22kHzAuto	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbPower	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.lnbVoltage	CHOICE	SAVE SETUP	EnumRange (13V 18V)
config.lnbPolSwitch	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.loFreqHigh	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.loFreqLow	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.bandEdge	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
dvbmode	CHOICE	SAVE	EnumRange (DVBS DVBS2)
frequency	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT	SAVE	DoubleRange (0.000 .. 0.000)
modulation	CHOICE	SAVE	EnumRange (QPSK 8PSK 16APSK QAM16 QAM32 QAM64 QAM128 QAM256 QAMAUTO VSB8 VSB16)
fec	CHOICE	SAVE	EnumRange (AUTO 1/2 2/3 3/4 3/5 4/5 5/6 6/7 7/8 8/9 9/10 NONE)
polarization	CHOICE	SAVE	EnumRange (H V)
rolloff	CHOICE	SAVE	EnumRange (AUTO 20% 25% 35%)

name	*type*	*flags*	*range*
flags.signal	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.carrier	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.feclock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.sync	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rxlock	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.level	FLOAT	R/O	DoubleRange R/O (0 .. 0)
state.ber	FLOAT	R/O	DoubleRange R/O (0.0E+00 .. 0.0E+00)
internal.f	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.t	TEXT	R/O	StringRange R/O
internal.v	TEXT	R/O	StringRange R/O
internal.stat	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RX Lock)

9.698 Vaisala-WXT

This page describes the device driver and the device window for a Vaisala WXT weather station. The driver is intended for monitoring only purpose.

It is required that the permanent output of data telegrams is disabled and the used protocol is set to ASCII. The driver simply reads wind speed and direction every seconds and includes limit/threshold checking.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Measurements
- --- The faults page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

The Vaisala WXT device needs some configuration parameters as well as the correct settings for the serial interface

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the device. The recommended line setting for the use with the sat-nms software is *19200,N,8,1* .
northOffset	Defines the deviation angle in degrees of the devices mounting position .
wind.hiThreshold	Defines the high wind limit in m/s which triggers the high wind speed fault.
temp.hiThreshold	Defines the high temperature limit in °Celsius which triggers the high temperature fault.
temp.lothreshold	Defines the low temperature limit in °Celsius which triggers the high wind speed fault.
counterResetMode	Defines precipation counter reset mode. **Manual:** The counter is reset with reset command **Automatic:** The counts are reset after each precipitation message. Limit: The rain counter or hail counter is reset, when it reaches the predefined limit Immediate:The counts are reset immediately after receiving the command

Remote Control

- The default address is 0
- The default baud rate is: 19200/8/N/1.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.voltage.ps	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage.3.5	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voltage.heat	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.heat	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.int	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
config.northOffset	INTEGER	SETUP	IntegerRange (-180 .. 180)
config.wind.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 60.0)
config.temp.hiThreshold	FLOAT	SAVE SETUP	DoubleRange (-52.0 .. 60.0)
config.temp.loThreshold	FLOAT	SAVE SETUP	DoubleRange (-52.0 .. 60.0)
config.counterResetMode	CHOICE	SETUP	EnumRange (MANUAL AUTOMATIC LIMIT IMMEDIATE)
speed.min	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
speed2.min	FLOAT	R/O	DoubleRange R/O (0.0 .. 220.0)
speed.max	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
speed2.max	FLOAT	R/O	DoubleRange R/O (0.0 .. 220.0)
speed.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 60.0)
speed2.average	FLOAT	R/O	DoubleRange R/O (0.0 .. 220.0)
direction.min	INTEGER	R/O	IntegerRange R/O (0 .. 360)
direction.max	INTEGER	R/O	IntegerRange R/O (0 .. 360)
direction.average	INTEGER	R/O	IntegerRange R/O (0 .. 360)

name	*type*	*flags*	*range*
windRose	TEXT	R/O	StringRange R/O
pressure	INTEGER	R/O	IntegerRange R/O (600 .. 1100)
pressure2	FLOAT	R/O	DoubleRange R/O (0.600 .. 1.100)
temp	FLOAT	R/O	DoubleRange R/O (-52.0 .. 60.0)
temp2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
humidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 100.0)
rain.acc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rain.dur.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
rain.dur.min	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rain.int	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hail.acc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hail.dur.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hail.dur.min	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hail.int	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
heating	CHOICE	R/O	EnumRange R/O (ON OFF)
rain.int.peak	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
hail.int.peak	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
counterReset	TEXT		StringRange
reset	TEXT		StringRange
internal.var	CHOICE	R/O	EnumRange R/O (MANUAL AUTOMATIC LIMIT IMMEDIATE)
internal.var2	TEXT	R/O	StringRange R/O
internal.var3	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (High Wind Speed)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low Temperature)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (High Temperature)

9.699 Vertex-7134-Step-Track

This page describes the device driver and the device window for the Vertex 7134 step track system.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and permits to set new pointing angles. Unlike other sat-nms drivers for antenna controllers, the Vertex 7134 provides a true jog mode with two configurable speeds. **The antenna will move until you click STOP once you clicked one of the blue arrow buttons !**
- --- The second parameter page displays / configures the tracking parameters.
- --- This page gives access to the 10 internal target memories of the step track controller. Clicking the diskette symbol on a line stores the actual target position to this memory. Clicking a blue arrow symbol moves the antenna to the displayed position and starts tracking there.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	*description*
address	The Vertex 7134 step track system does not use this parameter.
slowJogSpeed	configures the slow jog speed (single arrow buttons for AZ/EL).
fastJogSpeed	configures the slow jog speed (double arrow buttons for AZ/EL).
beaconInTargets	If this is set to yes, the beacon selection 1-4 is stored together with the antenna pointing for each target in memory. The target list shows the beacon selection. The Vertex 7134 however does **not** automatically apply the beacon selection with a GOTO command (clicking a blue arrow on the target memory page)

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The Vertex step track expects *1200,E,7,1* .

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.slowJogSpeed	FLOAT	SAVE SETUP	DoubleRange (0.1 .. 9.9)
config.fastJogSpeed	FLOAT	SAVE SETUP	DoubleRange (0.1 .. 9.9)
config.beaconInTargets	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.azLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.plLoLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
internal.remote	TEXT	R/O	StringRange R/O
internal.mva	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mvp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mve	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.faults	TEXT	R/O	StringRange R/O
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
dest.az	FLOAT		DoubleRange (0.00 .. 0.00)
dest.el	FLOAT		DoubleRange (0.00 .. 0.00)
dest.pl	FLOAT		DoubleRange (0.00 .. 0.00)
moveToDest	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.limit	BOOLEAN	R/O	BooleanRange R/O (true false)
stop	TEXT	NOPRESET	StringRange
trackImmediate	TEXT	NOPRESET	StringRange
jog	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
beaconSelect	CHOICE		EnumRange (BEACON 1 BEACON 2 BEACON 3 BEACON 4)
stepCycle	INTEGER		IntegerRange (0 .. 240)
gotoTarget	INTEGER	NOPRESET	IntegerRange (1 .. 10)
storeTarget	INTEGER	NOPRESET	IntegerRange (1 .. 10)
target.01.info	TEXT	R/O	StringRange R/O
target.02.info	TEXT	R/O	StringRange R/O
target.03.info	TEXT	R/O	StringRange R/O
target.04.info	TEXT	R/O	StringRange R/O
target.05.info	TEXT	R/O	StringRange R/O
target.06.info	TEXT	R/O	StringRange R/O
target.07.info	TEXT	R/O	StringRange R/O
target.08.info	TEXT	R/O	StringRange R/O
target.09.info	TEXT	R/O	StringRange R/O
target.10.info	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Out door power)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Hard limit)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Azimuth)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Elevation)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Maintenance override)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Emergency stop)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Step track)

9.700 Vertex-7200-Step-Track-DeltaPol

This page describes the device driver and the device window for the Vertex 7200 step track system. As of version 2.00, the driver is capable to control the essential settings of the ACU like antenna pointing or tracking start/stop. This version of the driver is able to handle units with a 4th axis (so call Delta Polarization).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the tracking mode / state and allows to start / stop resume tracking. The beacon receiver related parameters also are located at this page. Refer to the Vertex manual for details of the tracking modes in particular.
- .. --- On five pages you find the 50 target memories of the ACU. Clicking on the blue arrow next to the target name recalls the target position and tracking mode.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
address	The Vertex 7200 step track system does not use this parameter.
beaconReceiver	Configures if the ACU uses an TRL beacon receiver (TRL/253) or a third party model (OTHER). The setting other disables all beacon receiver related parameters other than the beacon level.
operatorPassword	The password for the operator security level must be set to the same value than configured at the ACU front panel. If the ACU has not configured a password, the values entered here does not matter.
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The Vertex step track expects *9600,N,8,1* .

Remarks

The Vertex 7200 restricts the control of the device to one of the M&C ports at a time. The sat-nms driver operates the 7200 ACU at 'operator' security level. To make the driver work properly it is important to

- configure all other M&C ports and the front panel to 'monitor' level.
- configure the 'operator' level password at the ACU and the sat-nms device setup to the same value.
- return the front panel security level back to 'monitor' after you (temporarily) operated the device from there.

If the Vertex 7200 has been configured to use soft limits, the device driver reads the limit values and modifies the valid range of the azimuth, elevation and polarization parameters accordingly. The limits are read once when the communication to ACU gets established. Hence, after changing the limit settings at the front panel, set the sat-nms device once to OUT-OF-SERVICE and back to OPERATIONAL to make it read the limits again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.rc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.r2	HEX	R/O	HexRange R/O (0 .. 0)
config.beaconReceiver	CHOICE	SAVE SETUP	EnumRange (TRL/253 OTHER)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.supervisorPasswd	TEXT	SAVE SETUP	StringRange
config.securityLevel	CHOICE	SAVE SETUP	EnumRange (MONITOR OPERATOR SUPERVISOR)
config.hasSwitchBoard	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.firmware	CHOICE	SAVE SETUP	EnumRange (<3.26.06 >3.26.06)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.deltaPlLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.deltaPlHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
deltaPol	FLOAT		DoubleRange (0.0 .. 0.0)
frequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
attenuation	FLOAT		DoubleRange DIS (0.0 .. 50.0)
input	CHOICE		EnumRange DIS (1 2)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.deltaPl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.target	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
rx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
tx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
rx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
trackImmediate	TEXT	NOPRESET	StringRange
trackStandby	TEXT	NOPRESET	StringRange
trackResume	TEXT	NOPRESET	StringRange
goto	INTEGER	NOPRESET	IntegerRange (1 .. 50)
targetName.1	TEXT	R/O	StringRange R/O
targetName.2	TEXT	R/O	StringRange R/O
targetName.3	TEXT	R/O	StringRange R/O
targetName.4	TEXT	R/O	StringRange R/O
targetName.5	TEXT	R/O	StringRange R/O
targetName.6	TEXT	R/O	StringRange R/O
targetName.7	TEXT	R/O	StringRange R/O
targetName.8	TEXT	R/O	StringRange R/O
targetName.9	TEXT	R/O	StringRange R/O
targetName.10	TEXT	R/O	StringRange R/O
targetName.11	TEXT	R/O	StringRange R/O
targetName.12	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
targetName.13	TEXT	R/O	StringRange R/O
targetName.14	TEXT	R/O	StringRange R/O
targetName.15	TEXT	R/O	StringRange R/O
targetName.16	TEXT	R/O	StringRange R/O
targetName.17	TEXT	R/O	StringRange R/O
targetName.18	TEXT	R/O	StringRange R/O
targetName.19	TEXT	R/O	StringRange R/O
targetName.20	TEXT	R/O	StringRange R/O
targetName.21	TEXT	R/O	StringRange R/O
targetName.22	TEXT	R/O	StringRange R/O
targetName.23	TEXT	R/O	StringRange R/O
targetName.24	TEXT	R/O	StringRange R/O
targetName.25	TEXT	R/O	StringRange R/O
targetName.26	TEXT	R/O	StringRange R/O
targetName.27	TEXT	R/O	StringRange R/O
targetName.28	TEXT	R/O	StringRange R/O
targetName.29	TEXT	R/O	StringRange R/O
targetName.30	TEXT	R/O	StringRange R/O
targetName.31	TEXT	R/O	StringRange R/O
targetName.32	TEXT	R/O	StringRange R/O
targetName.33	TEXT	R/O	StringRange R/O
targetName.34	TEXT	R/O	StringRange R/O
targetName.35	TEXT	R/O	StringRange R/O
targetName.36	TEXT	R/O	StringRange R/O
targetName.37	TEXT	R/O	StringRange R/O
targetName.38	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
targetName.39	TEXT	R/O	StringRange R/O
targetName.40	TEXT	R/O	StringRange R/O
targetName.41	TEXT	R/O	StringRange R/O
targetName.42	TEXT	R/O	StringRange R/O
targetName.43	TEXT	R/O	StringRange R/O
targetName.44	TEXT	R/O	StringRange R/O
targetName.45	TEXT	R/O	StringRange R/O
targetName.46	TEXT	R/O	StringRange R/O
targetName.47	TEXT	R/O	StringRange R/O
targetName.48	TEXT	R/O	StringRange R/O
targetName.49	TEXT	R/O	StringRange R/O
targetName.50	TEXT	R/O	StringRange R/O
internal.a	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.e	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.p	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ph	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.d	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.l	TEXT	R/O	StringRange R/O
internal.f	TEXT	R/O	StringRange R/O
internal.rxPol	TEXT	R/O	StringRange R/O
internal.txPol	TEXT	R/O	StringRange R/O
internal.lin	TEXT	R/O	StringRange R/O
internal.sw	TEXT	R/O	StringRange R/O
internal.sw1	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.sw2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw4	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw5	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw6	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.swflt	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Drive cabinet)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Motion)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.04	ALARM	R/O	AlarmFlagRange R/O (System)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch remote access)

9.701 Vertex-7200-Step-Track-Prec

This page describes the device driver and the device window for the Vertex 7200 step track system. As of version 2.00, the driver is capable to control the essential settings of the ACU like antenna pointing or tracking start/stop. This version of the driver is able to handle units with high precision angles.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the tracking mode / state and allows to start / stop resume tracking. The beacon receiver related parameters also are located at this page. Refer to the Vertex manual for details of the tracking modes in particular.
- .. --- On five pages you find the 50 target memories of the ACU. Clicking on the blue arrow next to the target name recalls the target position and tracking mode.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
address	The Vertex 7200 step track system does not use this parameter.
beaconReceiver	Configures if the ACU uses an TRL beacon receiver (TRL/253) or a third party model (OTHER). The setting other disables all beacon receiver related parameters other than the beacon level.
operatorPassword	The password for the operator security level must be set to the same value than configured at the ACU front panel. If the ACU has not configured a password, the values entered here does not matter.
securityLevel	Switch the driver between operation or monitoring mode.
hasPolarization	Define if a polarisation drive is equipped.
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The Vertex step track expects *9600,N,8,1* .

Remarks

The Vertex 7200 restricts the control of the device to one of the M&C ports at a time. The sat-nms driver operates the 7200 ACU at 'operator' security level. To make the driver work properly it is important to

- configure all other M&C ports and the front panel to 'monitor' level.
- configure the 'operator' level password at the ACU and the sat-nms device setup to the same value.
- return the front panel security level back to 'monitor' after you (temporarily) operated the device from there.

If the Vertex 7200 has been configured to use soft limits, the device driver reads the limit values and modifies the valid range of the azimuth, elevation and polarization parameters accordingly. The limits are read once when the communication to ACU gets established. Hence, after changing the limit settings at the front panel, set the sat-nms device once to OUT-OF-SERVICE and back to OPERATIONAL to make it read the limits again.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.rc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.r2	HEX	R/O	HexRange R/O (0 .. 0)
config.beaconReceiver	CHOICE	SAVE SETUP	EnumRange (TRL/253 OTHER)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.supervisorPasswd	TEXT	SAVE SETUP	StringRange
config.securityLevel	CHOICE	SAVE SETUP	EnumRange (MONITOR OPERATOR SUPERVISOR)
config.hasPolarization	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasSwitchBoard	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.firmware	CHOICE	SAVE SETUP	EnumRange (<3.26.06 >3.26.06)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
frequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
attenuation	FLOAT		DoubleRange DIS (0.0 .. 50.0)
input	CHOICE		EnumRange DIS (1 2)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.target	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
rx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
tx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
rx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
trackImmediate	TEXT	NOPRESET	StringRange
trackStandby	TEXT	NOPRESET	StringRange
trackResume	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
goto	INTEGER	NOPRESET	IntegerRange (1 .. 50)
targetName.1	TEXT	R/O	StringRange R/O
targetName.2	TEXT	R/O	StringRange R/O
targetName.3	TEXT	R/O	StringRange R/O
targetName.4	TEXT	R/O	StringRange R/O
targetName.5	TEXT	R/O	StringRange R/O
targetName.6	TEXT	R/O	StringRange R/O
targetName.7	TEXT	R/O	StringRange R/O
targetName.8	TEXT	R/O	StringRange R/O
targetName.9	TEXT	R/O	StringRange R/O
targetName.10	TEXT	R/O	StringRange R/O
targetName.11	TEXT	R/O	StringRange R/O
targetName.12	TEXT	R/O	StringRange R/O
targetName.13	TEXT	R/O	StringRange R/O
targetName.14	TEXT	R/O	StringRange R/O
targetName.15	TEXT	R/O	StringRange R/O
targetName.16	TEXT	R/O	StringRange R/O
targetName.17	TEXT	R/O	StringRange R/O
targetName.18	TEXT	R/O	StringRange R/O
targetName.19	TEXT	R/O	StringRange R/O
targetName.20	TEXT	R/O	StringRange R/O
targetName.21	TEXT	R/O	StringRange R/O
targetName.22	TEXT	R/O	StringRange R/O
targetName.23	TEXT	R/O	StringRange R/O
targetName.24	TEXT	R/O	StringRange R/O
targetName.25	TEXT	R/O	StringRange R/O
targetName.26	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
targetName.27	TEXT	R/O	StringRange R/O
targetName.28	TEXT	R/O	StringRange R/O
targetName.29	TEXT	R/O	StringRange R/O
targetName.30	TEXT	R/O	StringRange R/O
targetName.31	TEXT	R/O	StringRange R/O
targetName.32	TEXT	R/O	StringRange R/O
targetName.33	TEXT	R/O	StringRange R/O
targetName.34	TEXT	R/O	StringRange R/O
targetName.35	TEXT	R/O	StringRange R/O
targetName.36	TEXT	R/O	StringRange R/O
targetName.37	TEXT	R/O	StringRange R/O
targetName.38	TEXT	R/O	StringRange R/O
targetName.39	TEXT	R/O	StringRange R/O
targetName.40	TEXT	R/O	StringRange R/O
targetName.41	TEXT	R/O	StringRange R/O
targetName.42	TEXT	R/O	StringRange R/O
targetName.43	TEXT	R/O	StringRange R/O
targetName.44	TEXT	R/O	StringRange R/O
targetName.45	TEXT	R/O	StringRange R/O
targetName.46	TEXT	R/O	StringRange R/O
targetName.47	TEXT	R/O	StringRange R/O
targetName.48	TEXT	R/O	StringRange R/O
targetName.49	TEXT	R/O	StringRange R/O
targetName.50	TEXT	R/O	StringRange R/O
internal.a	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.e	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
internal.p	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.ph	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.l	TEXT	R/O	StringRange R/O
internal.f	TEXT	R/O	StringRange R/O
internal.rxPol	TEXT	R/O	StringRange R/O
internal.txPol	TEXT	R/O	StringRange R/O
internal.lin	TEXT	R/O	StringRange R/O
internal.sw	TEXT	R/O	StringRange R/O
internal.sw1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw4	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw5	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw6	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.swflt	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Drive cabinet)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Motion)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.04	ALARM	R/O	AlarmFlagRange R/O (System)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch remote access)

9.702 Vertex-7200-Step-Track

This page describes the device driver and the device window for the Vertex 7200 step track system. As of version 2.00, the driver is capable to control the essential settings of the ACU like antenna pointing or tracking start/stop.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the tracking mode / state and allows to start / stop resume tracking. The beacon receiver related parameters also are located at this page. Refer to the Vertex manual for details of the tracking modes in particular.
- .. --- On five pages you find the 50 target memories of the ACU. Clicking on the blue arrow next to the target name recalls the target position and tracking mode.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
address	The Vertex 7200 step track system does not use this parameter.
beaconReceiver	Configures if the ACU uses an TRL beacon receiver (TRL/253) or a third party model (OTHER). The setting other disables all beacon receiver related parameters other than the beacon level.
operatorPassword	The password for the operator security level must be set to the same value than configured at the ACU front panel. If the ACU has not configured a password, the values entered here does not matter.
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The Vertex step track expects *9600,N,8,1* .
config.hasPolarization	Select if unit is equipped with a polarization axis
config.hasSwitchBoard	Select if unit is equipped with a switch board
config.firmware	Select if firmware is newer than 3.26.06. This enables the driver to use the R7 command.
config.azLoLimit	Define lower azimuth limit. The value is not stored on the ACU itself.

parameter	*description*
config.azHiLimit	Define upper azimuth limit. The value is not stored on the ACU itself.
config.elLoLimit	Define upper elevation limit. The value is not stored on the ACU itself.
config.elHiLimit	Define upper elevation limit. The value is not stored on the ACU itself.
config.plLoLimit	Define upper polarization limit. The value is not stored on the ACU itself.
config.plHiLimit	Define upper polarization limit. The value is not stored on the ACU itself.

Remarks

The Vertex 7200 restricts the control of the device to one of the M&C ports at a time. The sat-nms driver operates the 7200 ACU at 'operator' security level. To make the driver work properly it is important to

- configure all other M&C ports and the front panel to 'monitor' level.
- configure the 'operator' level password at the ACU and the sat-nms device setup to the same value.
- return the front panel security level back to 'monitor' after you (temporarily) operated the device from there.

If the Vertex 7200 has been configured to use soft limits, the device driver reads the limit values and modifies the valid range of the azimuth, elevation and polarization parameters accordingly. The limits are read once when the communication to ACU gets established. Hence, after changing the limit settings at the front panel, set the sat-nms device once to OUT-OF-SERVICE and back to OPERATIONAL to make it read the limits again.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
internal.rc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.r2	HEX	R/O	HexRange R/O (0 .. 0)
config.beaconReceiver	CHOICE	SAVE SETUP	EnumRange (TRL/253 OTHER)
config.operatorPasswd	TEXT	SAVE SETUP	StringRange
config.supervisorPasswd	TEXT	SAVE SETUP	StringRange
config.securityLevel	CHOICE	SAVE SETUP	EnumRange (MONITOR OPERATOR SUPERVISOR)
config.hasPolarization	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.hasSwitchBoard	CHOICE	SAVE SETUP	EnumRange (YES NO)
config.firmware	CHOICE	SAVE SETUP	EnumRange (<3.26.06 >3.26.06)
config.azLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.azHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.elHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.plLoLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.plHiLimit	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
azimuth	FLOAT		DoubleRange (0.00 .. 0.00)
elevation	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
polarization	FLOAT		DoubleRange (0.0 .. 0.0)
frequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
attenuation	FLOAT		DoubleRange DIS (0.0 .. 50.0)
input	CHOICE		EnumRange DIS (1 2)
updateDest	TEXT	NOPRESET	StringRange
state.az	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.el	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.target	TEXT	R/O	StringRange R/O
state.beacon	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
rx.polPlane	CHOICE		EnumRange DIS (X Y RHCP LHCP)
tx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
rx.clSwitch	CHOICE		EnumRange DIS (CP LP INVALID)
clrflt	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
clearTracking	TEXT	NOPRESET	StringRange
trackImmediate	TEXT	NOPRESET	StringRange
trackStandby	TEXT	NOPRESET	StringRange
trackResume	TEXT	NOPRESET	StringRange
selectTarget	INTEGER	NOPRESET	IntegerRange (1 .. 50)
controlTracking	CHOICE	NOPRESET	EnumRange (OFF ON)
goto	INTEGER	NOPRESET	IntegerRange (1 .. 50)
targetName.1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
targetName.2	TEXT	R/O	StringRange R/O
targetName.3	TEXT	R/O	StringRange R/O
targetName.4	TEXT	R/O	StringRange R/O
targetName.5	TEXT	R/O	StringRange R/O
targetName.6	TEXT	R/O	StringRange R/O
targetName.7	TEXT	R/O	StringRange R/O
targetName.8	TEXT	R/O	StringRange R/O
targetName.9	TEXT	R/O	StringRange R/O
targetName.10	TEXT	R/O	StringRange R/O
targetName.11	TEXT	R/O	StringRange R/O
targetName.12	TEXT	R/O	StringRange R/O
targetName.13	TEXT	R/O	StringRange R/O
targetName.14	TEXT	R/O	StringRange R/O
targetName.15	TEXT	R/O	StringRange R/O
targetName.16	TEXT	R/O	StringRange R/O
targetName.17	TEXT	R/O	StringRange R/O
targetName.18	TEXT	R/O	StringRange R/O
targetName.19	TEXT	R/O	StringRange R/O
targetName.20	TEXT	R/O	StringRange R/O
targetName.21	TEXT	R/O	StringRange R/O
targetName.22	TEXT	R/O	StringRange R/O
targetName.23	TEXT	R/O	StringRange R/O
targetName.24	TEXT	R/O	StringRange R/O
targetName.25	TEXT	R/O	StringRange R/O
targetName.26	TEXT	R/O	StringRange R/O
targetName.27	TEXT	R/O	StringRange R/O
targetName.28	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
targetName.29	TEXT	R/O	StringRange R/O
targetName.30	TEXT	R/O	StringRange R/O
targetName.31	TEXT	R/O	StringRange R/O
targetName.32	TEXT	R/O	StringRange R/O
targetName.33	TEXT	R/O	StringRange R/O
targetName.34	TEXT	R/O	StringRange R/O
targetName.35	TEXT	R/O	StringRange R/O
targetName.36	TEXT	R/O	StringRange R/O
targetName.37	TEXT	R/O	StringRange R/O
targetName.38	TEXT	R/O	StringRange R/O
targetName.39	TEXT	R/O	StringRange R/O
targetName.40	TEXT	R/O	StringRange R/O
targetName.41	TEXT	R/O	StringRange R/O
targetName.42	TEXT	R/O	StringRange R/O
targetName.43	TEXT	R/O	StringRange R/O
targetName.44	TEXT	R/O	StringRange R/O
targetName.45	TEXT	R/O	StringRange R/O
targetName.46	TEXT	R/O	StringRange R/O
targetName.47	TEXT	R/O	StringRange R/O
targetName.48	TEXT	R/O	StringRange R/O
targetName.49	TEXT	R/O	StringRange R/O
targetName.50	TEXT	R/O	StringRange R/O
internal.a	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.e	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.p	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
internal.ph	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.l	TEXT	R/O	StringRange R/O
internal.f	TEXT	R/O	StringRange R/O
internal.rxPol	TEXT	R/O	StringRange R/O
internal.txPol	TEXT	R/O	StringRange R/O
internal.lin	TEXT	R/O	StringRange R/O
internal.sw	TEXT	R/O	StringRange R/O
internal.sw1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw3	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw4	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw5	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw6	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.swflt	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Drive cabinet)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Motion)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Tracking)
faults.04	ALARM	R/O	AlarmFlagRange R/O (System)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Switch remote access)

9.703 Vertex-ACU5000

This page describes the device driver and the device window for the Vertex antenna controller ACU5000.

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter parge contains the ACU's target list..
- --- This page displays the steptrack parameters and state information for this mode.
- --- This page displays the TRU (tracking receiver) parameters and the state information for the receiver
- --- This page shows various measurement values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.0.123:8088** ". The ACU5000 listens at port 8080 for remote control connections, therefore this port number always must be specified.
ftp.user	The user name to login at the ACU via sftp for TLE/OEM file uploads
ftp.password	The password to login at the ACU via sftp for TLE/OEM file uploads
ftp.oemdir	The directory on the ACU where to save OEM files on the ACU
ftp.tledir	The directory on the ACU where to save TLE files on the ACU
tru.lofreq	The LO frequency of the tracking receiver controlled by the ACU (MHz)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.buildDate	TEXT	R/O	StringRange R/O
info.coreVersion	TEXT	R/O	StringRange R/O
info.bitmapVersion	TEXT	R/O	StringRange R/O
info.deviceVersion	TEXT	R/O	StringRange R/O
info.osVersion	TEXT	R/O	StringRange R/O
info.configuration	TEXT	R/O	StringRange R/O
info.protocolVersion	TEXT	R/O	StringRange R/O
config.ftp.user	TEXT	SAVE SETUP	StringRange
config.ftp.password	TEXT	SAVE SETUP	StringRange
config.ftp.oemdir	TEXT	SAVE SETUP	StringRange
config.ftp.tledir	TEXT	SAVE SETUP	StringRange
config.tru.lofreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
internal.dummy	TEXT	R/O	StringRange R/O
internal.init	TEXT	R/O	StringRange R/O
internal.reply	TEXT	R/O	StringRange R/O
internal.summary	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.freq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
internal.azimuth	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
internal.elevation	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.azPosError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.onposition.az	BOOLEAN	R/O	BooleanRange R/O (true false)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elPosError	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.onposition.el	BOOLEAN	R/O	BooleanRange R/O (true false)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.azVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.elVelocity	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.movingThreshold	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
targets.select	CHOICE		EnumRange (UNKNOWN)
targets.content	TEXT	R/O	StringRange R/O
targets.apply	TEXT	NOPRESET	StringRange
tle.target	CHOICE		EnumRange (?)

name	*type*	*flags*	*range*
oem.target	CHOICE		EnumRange (?)
tru.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tru.atten	FLOAT		DoubleRange (0 .. 0)
tru.power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tru.polarization	CHOICE		EnumRange (n/a RHCP LHCP)
tru.rfmode	CHOICE		EnumRange (DIGITAL ANALOG WIDEBAND n/a)
tru.filter	CHOICE		EnumRange (n/a 62.5 Hz 125 Hz 250 Hz 500 Hz 1 kHz 4 kHz 250 kHz)
tru.sweepWidth	INTEGER		IntegerRange (0 .. 0)
tru.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
tru.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
st.initialLongitude	FLOAT		DoubleRange (0.000 .. 0.000)
st.initialAzimuth	FLOAT		DoubleRange (0.000 .. 0.000)
st.initialElevation	FLOAT		DoubleRange (0.000 .. 0.000)
st.thresholdBandwidth	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
st.stepSize	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
st.idleTime	INTEGER	R/O	IntegerRange R/O (0 .. 0)
st.fading	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
st.firstAxis	CHOICE	R/O	EnumRange R/O (AZIMUTH ELEVATION)
st.subMode	CHOICE	R/O	EnumRange R/O (IDLE PEAKING-1ST-AXIS PEAKING-2ND-AXIS)
st.remainingIdleTime	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
st.beaconLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
temperature	FLOAT		DoubleRange (-100.0 .. 100.0)
humidity	FLOAT		DoubleRange (0.0 .. 100.0)
pressure	FLOAT		DoubleRange (0.0 .. 1100.0)

name	*type*	*flags*	*range*
motorStop	TEXT	NOPRESET	StringRange
presetMode	TEXT	NOPRESET	StringRange
tleMode	TEXT	NOPRESET	StringRange
stepMode	TEXT	NOPRESET	StringRange
oemMode	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
uploadTLE	TEXT	NOPRESET	StringRange
uploadOEM	TEXT	NOPRESET	StringRange
uploadSuccess	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
internal.az.min	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.az.max	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.el.min	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
internal.el.max	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (COM TcAdsMain)
faults.02	ALARM	R/O	AlarmFlagRange R/O (COM TcAdsPLC)
faults.03	ALARM	R/O	AlarmFlagRange R/O (COM TRU)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ Position)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ SoftLimitCwUp)
faults.06	ALARM	R/O	AlarmFlagRange R/O (AZ SoftLimitCcwDown)
faults.07	ALARM	R/O	AlarmFlagRange R/O (AZ HardLimitCwUp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (AZ HardLimitCcwDown)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (AZ Drive 1 Error)
faults.10	ALARM	R/O	AlarmFlagRange R/O (AZ Drive 2 Error)
faults.11	ALARM	R/O	AlarmFlagRange R/O (AZ Interlock Axis)
faults.12	ALARM	R/O	AlarmFlagRange R/O (AZ Interlock Summary)
faults.13	ALARM	R/O	AlarmFlagRange R/O (EL Position)
faults.14	ALARM	R/O	AlarmFlagRange R/O (EL SoftLimitCwUp)
faults.15	ALARM	R/O	AlarmFlagRange R/O (EL SoftLimitCcwDown)
faults.16	ALARM	R/O	AlarmFlagRange R/O (EL HardLimitCwUp)
faults.17	ALARM	R/O	AlarmFlagRange R/O (EL HardLimitCcwDown)
faults.18	ALARM	R/O	AlarmFlagRange R/O (EL Drive 1 Error)
faults.19	ALARM	R/O	AlarmFlagRange R/O (EL Drive 2 Error)
faults.20	ALARM	R/O	AlarmFlagRange R/O (EL Interlock Axis)
faults.21	ALARM	R/O	AlarmFlagRange R/O (EL Interlock Summary)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Time DifferenceHigh)
faults.23	ALARM	R/O	AlarmFlagRange R/O (EStop Cabinet)
faults.24	ALARM	R/O	AlarmFlagRange R/O (EStop MovingStructure)
faults.25	ALARM	R/O	AlarmFlagRange R/O (EStop PCU)
faults.26	ALARM	R/O	AlarmFlagRange R/O (EStop SafetyRelay)

name	*type*	*flags*	*range*
faults.27	ALARM	R/O	AlarmFlagRange R/O (EStop Summary)
faults.28	ALARM	R/O	AlarmFlagRange R/O (ILock CabinetOvertemp)
faults.29	ALARM	R/O	AlarmFlagRange R/O (ILock EqRoomStop)
faults.30	ALARM	R/O	AlarmFlagRange R/O (ILock HLimOverride)
faults.31	ALARM	R/O	AlarmFlagRange R/O (ILock Summary)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Breaker Power)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Breaker ServoDrives)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Breaker CabinetFan)
faults.35	ALARM	R/O	AlarmFlagRange R/O (RainBlower)
faults.36	ALARM	R/O	AlarmFlagRange R/O (ST NoBeaconLevel)
faults.37	ALARM	R/O	AlarmFlagRange R/O (TRU Temperature)
faults.38	ALARM	R/O	AlarmFlagRange R/O (TRU Track)
faults.39	ALARM	R/O	AlarmFlagRange R/O (TRU Summary)
faults.40	ALARM	R/O	AlarmFlagRange R/O (TRU Autophase)
faults.41	ALARM	R/O	AlarmFlagRange R/O (TRU IF Synth Lock)
faults.42	ALARM	R/O	AlarmFlagRange R/O (TRU IF Synth Lock)
faults.43	ALARM	R/O	AlarmFlagRange R/O (TRU Phasing)

name	*type*	*flags*	*range*
faults.44	ALARM	R/O	AlarmFlagRange R/O (ACU In Local Mode)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Air Condition Master)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Air Condition Slave)

9.704 Vertex-ACU8100

This page describes the device driver and the device window for the Vertex antenna controller ACU8100. The driver supports the OPTrack and steptrack modes of the antenna controller but none of the program track modes.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and lets you move the antenna to a particular pointing.
- --- The second parameter page displays the steptrack parameters and state information for this mode.
- --- The OPT parameter page displays the OPT parameters and state information for this mode.
- --- The third parameter page displays the MTR (tracking receiver) parameters and the state information for the receiver
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	*description*

parameter	*description*
address	This is the only parameter that must be set. Enter the antenna controller's IP address in 'dotted quad' notation here. Example: " **192.168.0.123:8080** ". The ACU8100 listens at port 8080 for remote control connections, therefore this port number always must be specified.

Remarks

To accept commanded antenna pointing angles, the ACU8100 requires the mode to be set to "Preset". In other modes, commanded angles are remembered by the ACU but not echoed to the sat-nms M&C. When switching to "Preset" mode, the previously commanded angles become active and the antenna moves to these angles.

This may be confusing, therefore the ACU8100 device window disables the fields for the destination angles unless the ACU is in "Preset" mode.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
internal.dummy	TEXT	R/O	StringRange R/O
azimuth	FLOAT		DoubleRange (0.0000 .. 0.0000)
internal.azimuth	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
elevation	FLOAT		DoubleRange (0.0000 .. 0.0000)
state.az	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)

name	*type*	*flags*	*range*
state.el	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
targets.select	CHOICE		EnumRange (UNKNOWN)
targets.content	TEXT	R/O	StringRange R/O
targets.apply	TEXT	NOPRESET	StringRange
st.initialLongitude	FLOAT		DoubleRange (0.000 .. 0.000)
st.initialAzimuth	FLOAT		DoubleRange (0.000 .. 0.000)
st.initialElevation	FLOAT		DoubleRange (0.000 .. 0.000)
st.initialMode	CHOICE		EnumRange (START-AT-CURRENT-POS START-AT-INITIAL-AZEL START-AT-INITIAL-LONG START-AT-LAST-POS)
st.thresholdBandwidth	FLOAT		DoubleRange (0.00 .. 0.00)
st.stepCycles	INTEGER		IntegerRange (0 .. 0)
st.stepSizeAz	FLOAT		DoubleRange (0.000 .. 0.000)
st.stepSizeEl	FLOAT		DoubleRange (0.000 .. 0.000)
st.idleTime	INTEGER		IntegerRange (0 .. 0)
st.fading	FLOAT		DoubleRange (0.00 .. 0.00)
st.beaconAlarm	FLOAT		DoubleRange (0.00 .. 0.00)
st.firstAxis	CHOICE		EnumRange (AZIMUTH ELEVATION)
st.subMode	CHOICE	R/O	EnumRange R/O (IDLE PEAKING-1ST-AXIS PEAKING-2ND-AXIS)
st.remainingIdleTime	FLOAT		DoubleRange (0.0 .. 0.0)
st.lowBeaconLevel	BOOLEAN	R/O	BooleanRange R/O (true false)
st.noBeaconLevel	BOOLEAN	R/O	BooleanRange R/O (true false)
st.beaconLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
opt.initialLongitude	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
opt.initialAzimuth	FLOAT		DoubleRange (0.000 .. 0.000)
opt.initialElevation	FLOAT		DoubleRange (0.000 .. 0.000)
opt.initialMode	CHOICE		EnumRange (START-AT-CURRENT-POS START-AT-INITIAL-AZEL START-AT-INITIAL-LONG START-AT-LAST-POS)
opt.thresholdBandwidth	FLOAT		DoubleRange (0.00 .. 0.00)
opt.stepCycles	INTEGER		IntegerRange (0 .. 0)
opt.stepSizeAz	FLOAT		DoubleRange (0.000 .. 0.000)
opt.stepSizeEl	FLOAT		DoubleRange (0.000 .. 0.000)
opt.idleTime	INTEGER		IntegerRange (0 .. 0)
opt.fading	FLOAT		DoubleRange (0.00 .. 0.00)
opt.beaconAlarm	FLOAT		DoubleRange (0.00 .. 0.00)
opt.firstAxis	CHOICE		EnumRange (AZIMUTH ELEVATION)
opt.subMode	CHOICE	R/O	EnumRange R/O (IDLE PEAKING-1ST-AXIS PEAKING-2ND-AXIS)
opt.remainingIdleTime	FLOAT		DoubleRange (0.0 .. 0.0)
opt.lowBeaconLevel	BOOLEAN	R/O	BooleanRange R/O (true false)
opt.noBeaconLevel	BOOLEAN	R/O	BooleanRange R/O (true false)
opt.beaconLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
opt.azCalc	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
opt.elCalc	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mtr.channelCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mtr.ifFrequency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mtr.freqOffset	INTEGER		IntegerRange (0 .. 0)
mtr.searchRange	INTEGER		IntegerRange (0 .. 0)
mtr.loopBandwidth	INTEGER		IntegerRange (0 .. 0)
mtr.ifBandwidth	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
mtr.thresholdLevelBw	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
mtr.antiSideBand	CHOICE	R/O	EnumRange R/O (YES NO)
mtr.integrationTime	INTEGER		IntegerRange (0 .. 0)
mtr.mode	CHOICE		EnumRange (COHERENT NON-COHERENT)
mtr.thresholdLevel	FLOAT		DoubleRange (0.00 .. 0.00)
mtr.phase1	FLOAT		DoubleRange (0.000 .. 0.000)
mtr.gain1x	FLOAT		DoubleRange (0.000 .. 0.000)
mtr.gain1y	FLOAT		DoubleRange (0.000 .. 0.000)
mtr.x1polarity	CHOICE		EnumRange (POSITIVE NEGATIVE)
mtr.y1polarity	CHOICE		EnumRange (POSITIVE NEGATIVE)
mtr.frequencyRef	CHOICE		EnumRange (INTERNAL EXTERNAL)
mtr.voltageOffsX	FLOAT		DoubleRange (0.000 .. 0.000)
mtr.voltageOffsY	FLOAT		DoubleRange (0.000 .. 0.000)
mtr.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
mtr.dx	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mtr.dy	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mtr.power1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
mtr.errorRate	INTEGER		IntegerRange (0 .. 0)
mtr.c2no1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
mtr.useC2noEst	CHOICE		EnumRange (YES NO)
mtr.c2noStartF	INTEGER		IntegerRange (0 .. 0)
mtr.c2noStopF	INTEGER		IntegerRange (0 .. 0)
mtr.powIntTime	INTEGER		IntegerRange (0 .. 0)
mtr.autoValid	CHOICE	R/O	EnumRange R/O (YES NO)
mtr.phasing	TEXT		StringRange
mp.activated	CHOICE	R/O	EnumRange R/O (YES NO)

name	*type*	*flags*	*range*
mp.lockOn	CHOICE	R/O	EnumRange R/O (YES NO)
mp.lockOnCounter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
mp.lockTest	CHOICE	R/O	EnumRange R/O (YES NO)
mp.azError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
mp.elError	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
mp.agc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
mp.azVoltage	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.elVoltage	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.agcVoltage	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
mp.rawLockOn	CHOICE	R/O	EnumRange R/O (YES NO)
motorStop	TEXT	NOPRESET	StringRange
presetMode	TEXT	NOPRESET	StringRange
steptrackMode	TEXT	NOPRESET	StringRange
optMode	TEXT	NOPRESET	StringRange
autoTrackMode	TEXT	NOPRESET	StringRange
updateDest	TEXT	NOPRESET	StringRange
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (E-Stop Inside Tower)
faults.02	ALARM	R/O	AlarmFlagRange R/O (E-Stop Low Stairs)
faults.03	ALARM	R/O	AlarmFlagRange R/O (E-Stop Alidade Platform)
faults.04	ALARM	R/O	AlarmFlagRange R/O (E-Stop Hub)
faults.05	ALARM	R/O	AlarmFlagRange R/O (E-Stop PCU)
faults.06	ALARM	R/O	AlarmFlagRange R/O (E-Stop Device)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Interlock Door Pedestal)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Interlock Door Hub)
faults.09	ALARM	R/O	AlarmFlagRange R/O (General Safe)
faults.10	ALARM	R/O	AlarmFlagRange R/O (General Power (latched))
faults.11	ALARM	R/O	AlarmFlagRange R/O (General RF inhibit)
faults.12	ALARM	R/O	AlarmFlagRange R/O (General Lightning protection)
faults.13	ALARM	R/O	AlarmFlagRange R/O (General Power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (General 24V Power)
faults.15	ALARM	R/O	AlarmFlagRange R/O (General Breaker)
faults.16	ALARM	R/O	AlarmFlagRange R/O (General Cabinet overtemp)
faults.17	ALARM	R/O	AlarmFlagRange R/O (General PLC: ACU interface)
faults.18	ALARM	R/O	AlarmFlagRange R/O (General Hub Overtemp)
faults.19	ALARM	R/O	AlarmFlagRange R/O (General PCU Mode)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Time)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Year)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Az Pre-Limit cw)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Az Pre-Limit ccw)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Az Soft-Prelimit cw)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Az Soft-Prelimit ccw)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Az Softlimit cw)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Az Softlimit ccw)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Az Final Limit)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Az Computer disabled)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Az Axis disabled)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Az Servo)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Az Motion)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Az Brake 1)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Az Brake 2)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Az Breaker)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Az Amplifier 1)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Az Amplifier 2)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Az Motor 1 overtemp)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Az Motor 2 overtemp)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Az Aux mode 1)

name	*type*	*flags*	*range*
faults.42	ALARM	R/O	AlarmFlagRange R/O (Az Aux mode 2)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Az Stop)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Az Hand crank inserted)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Az Overspeed)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Az Breakes released)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Az Stop at LCP)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Az Power on)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Az Reg. resistor 1 overtemp)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Az Amp. power cycle interlock)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Az Tacho)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Az ACU immobile)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Az Encoder)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Az CAN Amplifier 1)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Az CAN Amplifier 2)
faults.56	ALARM	R/O	AlarmFlagRange R/O (El Pre-Limit up)
faults.57	ALARM	R/O	AlarmFlagRange R/O (El Pre-Limit down)
faults.58	ALARM	R/O	AlarmFlagRange R/O (El Soft-Prelimit up)
faults.59	ALARM	R/O	AlarmFlagRange R/O (El Soft-Prelimit down)

name	*type*	*flags*	*range*
faults.60	ALARM	R/O	AlarmFlagRange R/O (El Softlimit up)
faults.61	ALARM	R/O	AlarmFlagRange R/O (El Softlimit down)
faults.62	ALARM	R/O	AlarmFlagRange R/O (El Final Limit)
faults.63	ALARM	R/O	AlarmFlagRange R/O (El Computer disabled)
faults.64	ALARM	R/O	AlarmFlagRange R/O (El Axis disabled)
faults.65	ALARM	R/O	AlarmFlagRange R/O (El Servo)
faults.66	ALARM	R/O	AlarmFlagRange R/O (El Motion)
faults.67	ALARM	R/O	AlarmFlagRange R/O (El Breaker)
faults.68	ALARM	R/O	AlarmFlagRange R/O (El Amplifier 1)
faults.69	ALARM	R/O	AlarmFlagRange R/O (El Motor 1 overtemp)
faults.70	ALARM	R/O	AlarmFlagRange R/O (El Stop)
faults.71	ALARM	R/O	AlarmFlagRange R/O (El Hand crank inserted)
faults.72	ALARM	R/O	AlarmFlagRange R/O (El Overspeed)
faults.73	ALARM	R/O	AlarmFlagRange R/O (El Stop at LCP)
faults.74	ALARM	R/O	AlarmFlagRange R/O (El Power on)
faults.75	ALARM	R/O	AlarmFlagRange R/O (El Reg. resistor 1 overtemp)
faults.76	ALARM	R/O	AlarmFlagRange R/O (El Amp. power cycle interlock)
faults.77	ALARM	R/O	AlarmFlagRange R/O (El Low Level)
faults.78	ALARM	R/O	AlarmFlagRange R/O (El Tacho)

name	*type*	*flags*	*range*
faults.79	ALARM	R/O	AlarmFlagRange R/O (El ACU Immobile)
faults.80	ALARM	R/O	AlarmFlagRange R/O (El Encoder failure)
faults.81	ALARM	R/O	AlarmFlagRange R/O (El CAN Amplifier 1)
faults.82	ALARM	R/O	AlarmFlagRange R/O (MTR High RF)
faults.83	ALARM	R/O	AlarmFlagRange R/O (MTR PS)
faults.84	ALARM	R/O	AlarmFlagRange R/O (MTR Frequency ref.)
faults.85	ALARM	R/O	AlarmFlagRange R/O (MTR Summary)
faults.90	ALARM	R/O	AlarmFlagRange R/O (Low Beacon)
faults.91	ALARM	R/O	AlarmFlagRange R/O (Beacon Lock)

9.705 Vertex-Digital-Tracking-Receiver

This page describes the device driver and the device window for the Vertex digital tracking receiver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the configuration parameter and the current status of the receiver. Refer to the Vertex manual for details of the tracking modes in particular.
- --- Lets you edit and select up to four presets of of the receiver.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the receiver.

parameter	*description*
address	The Vertex Digital Tracking Receiver does not use this parameter.
Serial	Remember to set the line parameters for the serial interface connected to the antenna controller. The Vertex receiver expects *19200,N,8,1* . The receiver may configured to 9600 baud on the front panel as well.

Communication protocol / Interface type

The Vertex Digital Tracking Receiver requires the Vertex-7200 protocol. There are 3 Remote-control ports at the rear panel of the receiver:

- Port 1 --- RS232 only
- Port 2 --- RS232 or RS485
- Port 3 --- RS485 only

Connect with a crossed cable (pins: 2-3, 3-2, 5-5) to port 1 and configure this port on the front as remote control port: Configs - COMM Parameters - Port 1. Select the baud rate and set M&C Shell as Protocol.

To switch the control mode from local to remote or vice versa press the Shift and the Remote (Help) button together.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.partNo	TEXT	R/O	StringRange R/O
info.serialNo	TEXT	R/O	StringRange R/O
info.release	TEXT	R/O	StringRange R/O
info.cpu	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.dsp	TEXT	R/O	StringRange R/O
info.mcu	TEXT	R/O	StringRange R/O
config.wideBand	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL-CONTROL READ-ONLY)
frequency	FLOAT		DoubleRange (950.000 .. 22000.000)
gain	FLOAT		DoubleRange (-50.0 .. 0.0)
bandwidth	CHOICE		EnumRange (16 kHz 32 kHz 62.5 kHz 125 kHz 250 kHz 500 kHz)
detection	CHOICE		EnumRange DIS (FFT SIGNAL FFT NOISE RMS POWER RMS DENS)
averaging	INTEGER		IntegerRange (1 .. 2000)
threshold	FLOAT		DoubleRange (-199.00 .. -1.00)
polarization	CHOICE		EnumRange (1 2)
power	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
beacon1.save	TEXT		StringRange
beacon1.restore	TEXT		StringRange
beacon1.name	TEXT	SAVE	StringRange
beacon2.save	TEXT		StringRange
beacon2.restore	TEXT		StringRange
beacon2.name	TEXT	SAVE	StringRange
beacon3.save	TEXT		StringRange
beacon3.restore	TEXT		StringRange
beacon3.name	TEXT	SAVE	StringRange
beacon4.save	TEXT		StringRange
beacon4.restore	TEXT		StringRange
beacon4.name	TEXT	SAVE	StringRange

name	*type*	*flags*	*range*
internal.tport	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.cport	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.faults	HEX	R/O	HexRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Low input signal)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input saturated)
faults.04	ALARM	R/O	AlarmFlagRange R/O (MCU link loss)
faults.05	ALARM	R/O	AlarmFlagRange R/O (DSP link loss)
faults.06	ALARM	R/O	AlarmFlagRange R/O (DSP data loss)
faults.07	ALARM	R/O	AlarmFlagRange R/O (SPU overflow)
faults.08	ALARM	R/O	AlarmFlagRange R/O (TBT link)
faults.09	ALARM	R/O	AlarmFlagRange R/O (TBT summary)
faults.10	ALARM	R/O	AlarmFlagRange R/O (TBT in local ctrl)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Out of band)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Band setup)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Band 1 BDC)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Band 2 BDC)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Band 3 BDC)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Band 4 BDC)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PLL1 lock)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PLL2 lock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.20	ALARM	R/O	AlarmFlagRange R/O (NVRAM)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Mute switch)
faults.22	ALARM	R/O	AlarmFlagRange R/O (SPU link lock)

9.706 Vertex-Downconverter-legacy

This page describes the device driver and the device window for the Vertex Ku BAND UCS/B series downconverter with serial interface and the legacy protocol version. It uses the CPI-STX protocol.

It was tested with:

- SCR 12000
  - SW releases: SCR 1.23 and 1.25
- DSC 10-004-2
  - SW releases: DCS 0.36, 1.05 and 1.07

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy parameters.
- --- load/Store units memory.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
muteControl	Switches mute control between NORMAL and AUTO

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.muteControl	CHOICE	SETUP	EnumRange (NORMAL AUTO)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 29.9)
rx.attn	FLOAT		DoubleRange (0.0 .. 29.9)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.outputLevel	FLOAT		DoubleRange (0.0 .. 0.0)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
mem.location	INTEGER		IntegerRange (0 .. 99)
mem.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
mem.attn	FLOAT		DoubleRange (0.0 .. 29.9)
mem.recall	TEXT	NOPRESET	StringRange
mem.set	TEXT	NOPRESET	StringRange
mem.view	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
mem.program	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (STANDBY ONLINE AUTO)
position	CHOICE	R/O	EnumRange R/O (A B)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External reference)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PLO lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Synthesizer lock)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal reference)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Star switch)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output level monitor)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.707 Vertex-RSC-1100

This page describes the device driver and the device window for the Vertex LNA/LNV RSC-1100 1:1 redundancy controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational upconverter parameters
- --- Meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *19200,8N1* .

Communication protocol / Interface type

This device driver requires the Vertex protocol. Please remember do define a address or use the factory default 0.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.unitType	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
protection	CHOICE		EnumRange (MANUAL AUTO)
pol1	CHOICE		EnumRange (MAIN STANDBY)
position	CHOICE		EnumRange (A B)
preference	CHOICE		EnumRange (NONE UNIT1 UNIT2)
revert	CHOICE		EnumRange (ENABLED DISABLED)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Unit current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote access)
internal.unitType	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pol1	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.708 Vertex-RSC-1200

This page describes the device driver and the device window for the Vertex LNA/LNV RSC-1200 1:2 redundancy controller.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational upconverter parameters
- --- Meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *19200,8N1* .

Communication protocol / Interface type

This device driver requires the Vertex protocol. Please remember do define a address or use the factory default 0.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitType	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.current.3	INTEGER	R/O	IntegerRange R/O (0 .. 1000)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
protection	CHOICE		EnumRange (MANUAL AUTO)
pol1	CHOICE		EnumRange (MAIN STANDBY)
pol2	CHOICE		EnumRange (MAIN STANDBY)
position	CHOICE	R/O	EnumRange R/O (3-STBY 1-STBY 2-STBY)
position.1	CHOICE		EnumRange (A B)
position.2	CHOICE		EnumRange (A B)
priority	CHOICE		EnumRange (NONE POL1 POL2)
revert	CHOICE		EnumRange (ENABLED DISABLED)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Unit 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Unit 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Unit 3 (Spare))
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Power Supply 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Controller)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Unit current)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Protection)
internal.unitType	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.state	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fault2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pol1	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pol2	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.08	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.709 Vertex-SCR-Downconverter-SNMP-legacy

This page describes the device driver and the device window for the Vertex SCR series upconverter with Ethernet interface in legacy mode (early firmware versions) without support for StarSwitch redundancy switching.

Tested with model:

- SCR14000BU

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- Not used with this model.
- --- Not used with this model.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (1.8 .. 31.8)
rx.attn	FLOAT		DoubleRange (1.8 .. 31.8)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
info.model	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.current	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.voltage	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.ethVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.model	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ref. module)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Front panel)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ext. ref.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temp. Cal.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Eth. DHCP)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Eth. com.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (STARswitch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Logic board)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temp. high)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temp. low)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Invalid freq.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX ON)

9.710 Vertex-SCR-Downconverter-SNMP

This page describes the preliminary device driver and the device window for the Vertex SCR series downconverter via SNMP.

Please Note: The driver is not fully finished.

Tested with model:

- SCR12000RD

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters of the Downconverter,
- --- The Measurement readings of the Downconverter
- --- The redundancy switching parameters (not implemneted with this unit)
- --- The faults page.
- --- The device info page.

- 🔧 --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	description
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section
extRef	Sets the reference frequency setting
minFreq	Defines the minimum rx frequency of the converter
maxFreq	Defines the maximum rx frequency of the converter
minGain	Defines the minimum available gain of the converter
maxGain	Defines the maximum available gain of the converter

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	description
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses `public` as a default for read and write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.ethVersion	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.uptime	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
config.minFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.maxFreq	FLOAT	SAVE SETUP	DoubleRange (0.000 .. 0.000)
config.minGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.maxGain	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
rx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
rx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
internal.d	TEXT	R/O	StringRange R/O
internal.h	TEXT	R/O	StringRange R/O
internal.m	TEXT	R/O	StringRange R/O
internal.s	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Ref. module)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Front panel)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ext. ref.)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Temp. Cal.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Eth. DHCP)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Eth. com.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (STARswitch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Logic board)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temp. high)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temp. low)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Power Up)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Invalid freq.)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX ON)

9.711 Vertex-SCR-Downconverter

This page describes the device driver and the device window for the Vertex SCR series downconverter with serial interface and support for the StarSwitch redundancy switching.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)

name	*type*	*flags*	*range*
location	CHOICE		EnumRange (STANDBY ONLINE AUTO)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
info.model	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.current	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.voltage	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.model	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.maxGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ref. module)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Front panel)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ext. ref.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temp. Cal.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Eth. DHCP)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Eth. com.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (STARswitch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Logic board)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temp. high)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temp. low)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Invalid freq.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX ON)

9.712 Vertex-SCR-Upconverter-SNMP-legacy

This page describes the device driver and the device window for the Vertex SCR series upconverter with Ethernet interface in legacy mode (early firmware versions) without support for StarSwitch redundancy switching.

Tested with model:

- SCR14000BU

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

- --- Not used with this model.
- --- Not used with this model.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- powerDisplayMode --- Defines the display of gain or attenuation in the parameter section.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (1.8 .. 31.8)
tx.attn	FLOAT		DoubleRange (1.8 .. 31.8)
tx.frequency	FLOAT		DoubleRange (13750.000 .. 14500.000)
info.model	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.current	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.voltage	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.ethVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
internal.model	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ref. module)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Front panel)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ext. ref.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temp. Cal.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Eth. DHCP)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Eth. com.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (STARswitch)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Logic board)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temp. high)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temp. low)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Invalid freq.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX ON)

9.713 Vertex-SCR-Upconverter

This page describes the device driver and the device window for the Vertex SCR series upconverter with serial interface and support for the StarSwitch redundancy switching.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL)
location	CHOICE		EnumRange (STANDBY ONLINE AUTO)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
info.model	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.current	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.voltage	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT	NOPRESET	StringRange
internal.model	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.minGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.maxGain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.minFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.maxFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Ref. module)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Front panel)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ext. ref.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temp. Cal.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Eth. DHCP)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Eth. com.)
faults.09	ALARM	R/O	AlarmFlagRange R/O (STARswitch)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (Logic board)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temp. high)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temp. low)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Invalid freq.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX ON)

9.714 Vertex-SSPA

This page describes the device driver and the device window for the Vertex SSPA amplifier SMR/DMR family. It was tested with in a 1+1 redundancy configuration. The drivers supports only units with one amplifier module included.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational upconverter parameters
- --- Meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
txPowerupState	Sets the power-up state of the SSPA.

parameter	*description*
powerUnit	Allows to select the unit of the power measurement in the front panel and the M&C protocol. This version of the driver needs "Watt" selected, because the parameter meas.fwdPwr is configured with Watt as unit.
powerMeasType	Allows to select the measurement for the reflected Power. You have to select "Power"
Serial	Remember to set the line parameters for the serial interface connected to the down-converter. It is recommended to operate the device at *19200,8N1* .

Communication protocol / Interface type

This device driver requires the Peak protocol. Please remember do define a address or use the factory default 0.

Installation

Before you can operate the down-converter with the satnms software, you must prepare the unit to match the interface parameters expected by the software.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.unitType	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.test	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.txPowerupState	CHOICE	SETUP	EnumRange (MUTED OPERATING PREVIOUS)
config.powerUnit	CHOICE	SETUP	EnumRange (dBm dBW Watt)
config.powerMeasType	CHOICE	SETUP	EnumRange (Power Return Loss (db) VSWR ratio)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (-20.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0 .. 0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.FETGateVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.inputVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.regulatedVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	FLOAT	R/O	DoubleRange R/O (0.0 .. 999.9)
buc.state	TEXT	R/O	StringRange R/O
red.mode	CHOICE		EnumRange (AUTO MANUAL)
red.switch	CHOICE		EnumRange (STANDBY ONLINE)
tx.switch	CHOICE		EnumRange (DUMMY ANTENNA)
internal.unitType	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sta3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt1	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.flt2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt9	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt10	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt11	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flt12	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.txon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.mode	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.txsw	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.redsw	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power Supply Disconn.)
faults.02	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Logic Board)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.07	ALARM	R/O	AlarmFlagRange R/O (SSPA module)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Fan 4)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Fan 3)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Fan 2)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 1)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Front panel)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Refl.Pwr Detector)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Maintenance Mode)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Block Upconverter)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Redundancy Comm)
faults.18	ALARM	R/O	AlarmFlagRange R/O (1+1 Switch Stuck)
faults.19	ALARM	R/O	AlarmFlagRange R/O (1+1 Switch Disconn.)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TX Switch Stuck)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TX Switch Disconn.)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.715 Vertex-TWTA-2100TK

This page describes the device driver and the device window for the Vertex RSI 2100TK TWT amplifiers. The driver has been tested with the following SW versions

- SW 4.1
- SW 4.3

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY)

- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). While communicating via RS422 the first HPA's address is 65 (Vertex=17) and the second HPA's address is 66 (Vertex=18).
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,E,7,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 34.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 34.0)
reset	TEXT	NOPRESET	StringRange
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.iHeater	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iBeam	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.plateTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.uBeam	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.coLowLim	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.coHigLim	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.coRefHigLim	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.txMeas	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PS Temperature)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Inverter)
faults.03	ALARM	R/O	AlarmFlagRange R/O (TWT Reflected RF)
faults.04	ALARM	R/O	AlarmFlagRange R/O (TWT Temperature)
faults.05	ALARM	R/O	AlarmFlagRange R/O (HVPS)
faults.06	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Reflected RF)
faults.08	ALARM	R/O	AlarmFlagRange R/O (High Input Power)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Low Input Power)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF (alarm))
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF (alarm))
faults.12	ALARM	R/O	AlarmFlagRange R/O (Plate Temperature)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Cabinet Temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Cathode voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Helix current)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Heater current)

name	*type*	*flags*	*range*
faults.17	ALARM	R/O	AlarmFlagRange R/O (Summary Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Standby Ready)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.716 Vertex-Upconverter-legacy

This page describes the device driver and the device window for the Vertex Ku BAND USC/B series upconverter with serial interface and the legacy protocol version. It uses the CPI-STX protocol.

It was tested with:

- USC 13-004-01
- SW releases: 1.05 and 1.06

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy parameters.
- --- load/Store units memory.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
muteControl	Switches mute control between NORMAL and AUTO

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.muteControl	CHOICE	SETUP	EnumRange (NORMAL AUTO)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.ifGain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.ifAttn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)

name	*type*	*flags*	*range*
tx.outputLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT	NOPRESET	StringRange
mem.location	INTEGER		IntegerRange (0 .. 99)
mem.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
mem.attn	FLOAT		DoubleRange (0.0 .. 30.0)
mem.ifAttn	FLOAT		DoubleRange (0.0 .. 30.0)
mem.recall	TEXT	NOPRESET	StringRange
mem.set	TEXT	NOPRESET	StringRange
mem.view	TEXT	NOPRESET	StringRange
mem.program	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (STANDBY ONLINE AUTO)
position	CHOICE	R/O	EnumRange R/O (A B)
internal.f1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.f2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External reference)
faults.04	ALARM	R/O	AlarmFlagRange R/O (PLO lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Synthesizer lock)

name	*type*	*flags*	*range*
faults.06	ALARM	R/O	AlarmFlagRange R/O (Internal reference)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Star switch)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output level monitor)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.717 ViaLite-HRA-Dual-EDFA

This page describes the device driver and the device window for the Dual output ViaLite Erbium Doped Fiber Amplifier

The driver was tested with the following models and SW:

- model: HRA-4-BB-(T-ZZ-D016-BH
- SW: 1.27 (2019-03-19)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements as well as their alarm state.
- --- Define all measurement alarm thresholds.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.

  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

- model --- Allows to define if the model has variable or fixed gain. Variable gain enables setting the gain control mode.

- noPumps --- Allows to define the number of pumps (2 or 4).

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.model	CHOICE	SAVE SETUP	EnumRange (FIXED GAIN VARIABLE GAIN)
config.noPumps	CHOICE	SAVE SETUP	EnumRange (2 4)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
input.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
input.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
input.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
input.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
output.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
output.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
input.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
input.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
input.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
input.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
output.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
output.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.3.loThreshold	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
current.3.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.3.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.3.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.4.loThreshold	INTEGER		IntegerRange (0 .. 0)
current.4.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.4.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.4.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.3.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.3.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.3.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.3.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.4.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.4.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.4.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.4.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
temp.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.3.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.3.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.3.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.3.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.4.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.4.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.4.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.4.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
meas.input.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.input.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
gain.1	FLOAT		DoubleRange (0.0 .. 25.0)
gain.2	FLOAT		DoubleRange (0.0 .. 25.0)
tx.requestedLevel.1	FLOAT		DoubleRange (0.0 .. 25.0)
tx.requestedLevel.2	FLOAT		DoubleRange (0.0 .. 25.0)
ctrl.mode.1	CHOICE	R/O	EnumRange R/O DIS (AUTO MANUAL GAIN POWER)

name	*type*	*flags*	*range*
ctrl.mode.2	CHOICE	R/O	EnumRange R/O DIS (AUTO MANUAL GAIN POWER)
loss	CHOICE		EnumRange DIS (ON OFF)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.3	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.temp.4	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
meas.tec.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.tec.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.tec.3	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
meas.tec.4	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.3	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
meas.current.4	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input 1 Low Warn)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input 1 Low)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input 1 High Warn)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input 1 High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output 1 Low Warn)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output 1 Low)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output 1 High Warn)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output 1 High)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Input 2 Low Warn)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Input 2 Low)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Input 2 High Warn)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Input 2 High)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Output 2 Low Warn)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Output 2 Low)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Output 2 High Warn)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Output 2 High)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. Low Warn)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. Low)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. High Warn)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. High)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Bias 1 Low Warn)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Bias 1 Low)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Bias 1 High Warn)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Bias 1 High)
faults.25	ALARM	R/O	AlarmFlagRange R/O (TEC 1 Low Warn)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TEC 1 Low)
faults.27	ALARM	R/O	AlarmFlagRange R/O (TEC 1 High Warn)

name	*type*	*flags*	*range*
faults.28	ALARM	R/O	AlarmFlagRange R/O (TEC 1 High)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Temp 1 Low Warn)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Temp 1 Low)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Temp 1 High Warn)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Temp 1 High)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Bias 2 Low Warn)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Bias 2 Low)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Bias 2 High Warn)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Bias 2 High)
faults.37	ALARM	R/O	AlarmFlagRange R/O (TEC 2 Low Warn)
faults.38	ALARM	R/O	AlarmFlagRange R/O (TEC 2 Low)
faults.39	ALARM	R/O	AlarmFlagRange R/O (TEC 2 High Warn)
faults.40	ALARM	R/O	AlarmFlagRange R/O (TEC 2 High)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Temp 2 Low Warn)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Temp 2 Low)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Temp 2 High Warn)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Temp 2 High)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Bias 3 Low Warn)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Bias 3 Low)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Bias 3 High Warn)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Bias 3 High)

name	*type*	*flags*	*range*
faults.49	ALARM	R/O	AlarmFlagRange R/O (TEC 3 Low Warn)
faults.50	ALARM	R/O	AlarmFlagRange R/O (TEC 3 Low)
faults.51	ALARM	R/O	AlarmFlagRange R/O (TEC 3 High Warn)
faults.52	ALARM	R/O	AlarmFlagRange R/O (TEC 3 High)
faults.53	ALARM	R/O	AlarmFlagRange R/O (Temp 3 Low Warn)
faults.54	ALARM	R/O	AlarmFlagRange R/O (Temp 3 Low)
faults.55	ALARM	R/O	AlarmFlagRange R/O (Temp 3 High Warn)
faults.56	ALARM	R/O	AlarmFlagRange R/O (Temp 3 High)
faults.57	ALARM	R/O	AlarmFlagRange R/O (Bias 4 Low Warn)
faults.58	ALARM	R/O	AlarmFlagRange R/O (Bias 4 Low)
faults.59	ALARM	R/O	AlarmFlagRange R/O (Bias 4 High Warn)
faults.60	ALARM	R/O	AlarmFlagRange R/O (Bias 4 High)
faults.61	ALARM	R/O	AlarmFlagRange R/O (TEC 4 Low Warn)
faults.62	ALARM	R/O	AlarmFlagRange R/O (TEC 4 Low)
faults.63	ALARM	R/O	AlarmFlagRange R/O (TEC 4 High Warn)
faults.64	ALARM	R/O	AlarmFlagRange R/O (TEC 4 High)
faults.65	ALARM	R/O	AlarmFlagRange R/O (Temp 4 Low Warn)
faults.66	ALARM	R/O	AlarmFlagRange R/O (Temp 4 Low)
faults.67	ALARM	R/O	AlarmFlagRange R/O (Temp 4 High Warn)
faults.68	ALARM	R/O	AlarmFlagRange R/O (Temp 4 High)
faults.69	ALARM	R/O	AlarmFlagRange R/O (DC 1 Low Warn)

name	*type*	*flags*	*range*
faults.70	ALARM	R/O	AlarmFlagRange R/O (DC 1 Low)
faults.71	ALARM	R/O	AlarmFlagRange R/O (DC 1 High Warn)
faults.72	ALARM	R/O	AlarmFlagRange R/O (DC 1 High)
faults.73	ALARM	R/O	AlarmFlagRange R/O (DC 2 Low Warn)
faults.74	ALARM	R/O	AlarmFlagRange R/O (DC 2 Low)
faults.75	ALARM	R/O	AlarmFlagRange R/O (DC 2 High Warn)
faults.76	ALARM	R/O	AlarmFlagRange R/O (DC 2 High)

9.718 ViaLite-HRA-EDFA

This page describes the device driver and the device window for the ViaLite Erbium Doped Fiber Amplifier

The driver was tested with the following models and SW:

- model: HRA-4-BB
- SW: 1.27 (2019-03-19)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements as well as their alarm state.
- --- Define all measurement alarm thresholds.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*

- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*

- *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
- *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
- *read-community* --- The SNMP read community name. This must match the read community expected by the device.
- *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.model	CHOICE	SAVE SETUP	EnumRange (FIXED GAIN VARIABLE GAIN)
config.minGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.maxGain	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.minLevel	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.maxLevel	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
input.loThreshold	INTEGER		IntegerRange (0 .. 0)
input.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
input.hiThreshold	INTEGER		IntegerRange (0 .. 0)
input.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.loThreshold	INTEGER		IntegerRange (0 .. 0)
output.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
output.hiThreshold	INTEGER		IntegerRange (0 .. 0)
output.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
current.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
tec.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
tec.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
meas.input	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.output	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
gain	FLOAT		DoubleRange (13.0 .. 22.0)
tx.requestedLevel	FLOAT		DoubleRange (17.5 .. 23.0)
ctrl.mode	CHOICE	R/O	EnumRange R/O DIS (AUTO MANUAL GAIN POWER)
loss	CHOICE		EnumRange DIS (ON OFF)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.tec.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
meas.tec.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.current.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.current.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input Low Warn)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input Low)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input High Warn)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Output Low Warn)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Output Low)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Output High Warn)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Output High)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. Low Warn)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. Low)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. High Warn)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit Temp. High)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Bias 1 Low Warn)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Bias 1 Low)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Bias 1 High Warn)

name	*type*	*flags*	*range*
faults.16	ALARM	R/O	AlarmFlagRange R/O (Bias 1 High)
faults.17	ALARM	R/O	AlarmFlagRange R/O (TEC 1 Low Warn)
faults.18	ALARM	R/O	AlarmFlagRange R/O (TEC 1 Low)
faults.19	ALARM	R/O	AlarmFlagRange R/O (TEC 1 High Warn)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TEC 1 High)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Temp 1 Low Warn)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Temp 1 Low)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Temp 1 High Warn)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Temp 1 High)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Bias 2 Low Warn)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Bias 2 Low)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Bias 2 High Warn)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Bias 2 High)
faults.29	ALARM	R/O	AlarmFlagRange R/O (TEC 2 Low Warn)
faults.30	ALARM	R/O	AlarmFlagRange R/O (TEC 2 Low)
faults.31	ALARM	R/O	AlarmFlagRange R/O (TEC 2 High Warn)
faults.32	ALARM	R/O	AlarmFlagRange R/O (TEC 2 High)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Temp 2 Low Warn)
faults.34	ALARM	R/O	AlarmFlagRange R/O (Temp 2 Low)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Temp 2 High Warn)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (Temp 2 High)
faults.37	ALARM	R/O	AlarmFlagRange R/O (DC 1 Low Warn)
faults.38	ALARM	R/O	AlarmFlagRange R/O (DC 1 Low)
faults.39	ALARM	R/O	AlarmFlagRange R/O (DC 1 High Warn)
faults.40	ALARM	R/O	AlarmFlagRange R/O (DC 1 High)
faults.41	ALARM	R/O	AlarmFlagRange R/O (DC 2 Low Warn)
faults.42	ALARM	R/O	AlarmFlagRange R/O (DC 2 Low)
faults.43	ALARM	R/O	AlarmFlagRange R/O (DC 2 High Warn)
faults.44	ALARM	R/O	AlarmFlagRange R/O (DC 2 High)

9.719 ViaLite-HRS-Switch

This page describes the device driver and the device window for the ViaLite HRS optical switch series

The driver was tested with the following models and SW:

- Models: HRS-5-0B-8T-01
- SW: v 1.2410

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all measurements as well as their alarm state.
- --- Define all measurement alarm thresholds.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- readBackDelay --- Delays the read back of a parameter actually set for at least the given delay time. This shall ensure that the device has processed the new parameter before it is read back. You should leave this parameter at it's default setting.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
input.a.loThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.a.loCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.a.hiThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.a.hiCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.b.loThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.b.loCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.b.hiThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
input.b.hiCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
output.loThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
output.loCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
output.hiThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
output.hiCritThreshold	FLOAT		DoubleRange (0.0 .. 0.0)
temp.loThreshold	INTEGER		IntegerRange (0 .. 0)
temp.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiThreshold	INTEGER		IntegerRange (0 .. 0)
temp.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.1.loThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.1.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.1.hiThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.1.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.2.loThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.2.loCritThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.2.hiThreshold	INTEGER		IntegerRange (0 .. 0)
voltage.2.hiCritThreshold	INTEGER		IntegerRange (0 .. 0)
internal.pos	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
meas.input.a	FLOAT		DoubleRange (0.0 .. 0.0)
meas.input.b	FLOAT		DoubleRange (0.0 .. 0.0)
meas.output	FLOAT		DoubleRange (0.0 .. 0.0)
protection	CHOICE		EnumRange (AUTO MANUAL)
position	CHOICE		EnumRange (A B)
switchover	FLOAT		DoubleRange (0.0 .. 0.0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.voltage.1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage.2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input A Low Warn)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Input A Low)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Input A High Warn)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Input A High)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Input B Low Warn)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Input B Low)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Input B High Warn)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Input B High)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Output Low Warn)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Output Low)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Output High Warn)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Output High)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temp. Low Warn)

name	*type*	*flags*	*range*
faults.14	ALARM	R/O	AlarmFlagRange R/O (Temp. Low)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Temp. High Warn)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Temp. High)
faults.17	ALARM	R/O	AlarmFlagRange R/O (+5VDC Low Warn)
faults.18	ALARM	R/O	AlarmFlagRange R/O (+5VDC Low)
faults.19	ALARM	R/O	AlarmFlagRange R/O (+5VDC High Warn)
faults.20	ALARM	R/O	AlarmFlagRange R/O (+5VDC High)
faults.21	ALARM	R/O	AlarmFlagRange R/O (-5VDC Low Warn)
faults.22	ALARM	R/O	AlarmFlagRange R/O (-5VDC Low)
faults.23	ALARM	R/O	AlarmFlagRange R/O (-5VDC High Warn)
faults.24	ALARM	R/O	AlarmFlagRange R/O (-5VDC High)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Protection)

9.720 ViaLiteHD-FO

This page describes the device driver and the device window for the ViaLite HD Optical link chassis with up 13 cards. It supports RX, TX DTX and DRX cards

The driver recognizes the number and type of installed cards and sets their ranges and variables accordingly

The driver was tested with the following cards and SW release:

- Cards: DRX, TX-DWDM
- SW: 1.0.9

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows all sensor measurements as well as their thresholds

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- highTempThreshold --- Defines the high temperature threshold in °C.
- lowVoltThreshold --- Defines the low voltage threshold in V.
- highVoltThreshold --- Defines the high voltage threshold in V.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.macAddress	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.voltage	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.slot.01.serial	TEXT	R/O	StringRange R/O
info.slot.02.serial	TEXT	R/O	StringRange R/O
info.slot.03.serial	TEXT	R/O	StringRange R/O
info.slot.04.serial	TEXT	R/O	StringRange R/O
info.slot.05.serial	TEXT	R/O	StringRange R/O
info.slot.06.serial	TEXT	R/O	StringRange R/O
info.slot.07.serial	TEXT	R/O	StringRange R/O
info.slot.08.serial	TEXT	R/O	StringRange R/O
info.slot.09.serial	TEXT	R/O	StringRange R/O
info.slot.10.serial	TEXT	R/O	StringRange R/O
info.slot.11.serial	TEXT	R/O	StringRange R/O
info.slot.12.serial	TEXT	R/O	StringRange R/O
info.slot.13.serial	TEXT	R/O	StringRange R/O
config.highTempThreshold	INTEGER	SETUP	IntegerRange (0 .. 90)
config.lowVoltThreshold	FLOAT	SETUP	DoubleRange (9.0 .. 15.0)
config.highVoltThreshold	FLOAT	SETUP	DoubleRange (9.0 .. 15.0)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.1	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O
internal.flt	TEXT	R/O	StringRange R/O
internal.01.card	TEXT	R/O	StringRange R/O
internal.02.card	TEXT	R/O	StringRange R/O
internal.03.card	TEXT	R/O	StringRange R/O
internal.04.card	TEXT	R/O	StringRange R/O
internal.05.card	TEXT	R/O	StringRange R/O
internal.06.card	TEXT	R/O	StringRange R/O
internal.07.card	TEXT	R/O	StringRange R/O
internal.08.card	TEXT	R/O	StringRange R/O
internal.09.card	TEXT	R/O	StringRange R/O
internal.10.card	TEXT	R/O	StringRange R/O
internal.11.card	TEXT	R/O	StringRange R/O
internal.12.card	TEXT	R/O	StringRange R/O
internal.13.card	TEXT	R/O	StringRange R/O
internal.01.actCard	TEXT	R/O	StringRange R/O
internal.02.actCard	TEXT	R/O	StringRange R/O
internal.03.actCard	TEXT	R/O	StringRange R/O
internal.04.actCard	TEXT	R/O	StringRange R/O
internal.05.actCard	TEXT	R/O	StringRange R/O
internal.06.actCard	TEXT	R/O	StringRange R/O
internal.07.actCard	TEXT	R/O	StringRange R/O
internal.08.actCard	TEXT	R/O	StringRange R/O
internal.09.actCard	TEXT	R/O	StringRange R/O
internal.10.actCard	TEXT	R/O	StringRange R/O
internal.11.actCard	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.12.actCard	TEXT	R/O	StringRange R/O
internal.13.actCard	TEXT	R/O	StringRange R/O
internal.01.direction	TEXT	R/O	StringRange R/O
internal.02.direction	TEXT	R/O	StringRange R/O
internal.03.direction	TEXT	R/O	StringRange R/O
internal.04.direction	TEXT	R/O	StringRange R/O
internal.05.direction	TEXT	R/O	StringRange R/O
internal.06.direction	TEXT	R/O	StringRange R/O
internal.07.direction	TEXT	R/O	StringRange R/O
internal.08.direction	TEXT	R/O	StringRange R/O
internal.09.direction	TEXT	R/O	StringRange R/O
internal.10.direction	TEXT	R/O	StringRange R/O
internal.11.direction	TEXT	R/O	StringRange R/O
internal.12.direction	TEXT	R/O	StringRange R/O
internal.13.direction	TEXT	R/O	StringRange R/O
slot.01.version	TEXT	R/O	StringRange R/O
slot.01.model	TEXT	R/O	StringRange R/O
slot.01.name	TEXT		StringRange
slot.01.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.01.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.02.version	TEXT	R/O	StringRange R/O
slot.02.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot.02.name	TEXT		StringRange
slot.02.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.02.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.03.version	TEXT	R/O	StringRange R/O
slot.03.model	TEXT	R/O	StringRange R/O
slot.03.name	TEXT		StringRange
slot.03.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.03.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.04.version	TEXT	R/O	StringRange R/O
slot.04.model	TEXT	R/O	StringRange R/O
slot.04.name	TEXT		StringRange
slot.04.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.04.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.05.version	TEXT	R/O	StringRange R/O
slot.05.model	TEXT	R/O	StringRange R/O
slot.05.name	TEXT		StringRange

name	*type*	*flags*	*range*
slot.05.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.05.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.06.version	TEXT	R/O	StringRange R/O
slot.06.model	TEXT	R/O	StringRange R/O
slot.06.name	TEXT		StringRange
slot.06.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.06.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.07.version	TEXT	R/O	StringRange R/O
slot.07.model	TEXT	R/O	StringRange R/O
slot.07.name	TEXT		StringRange
slot.07.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.07.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.08.version	TEXT	R/O	StringRange R/O
slot.08.model	TEXT	R/O	StringRange R/O
slot.08.name	TEXT		StringRange

name	*type*	*flags*	*range*
slot.08.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.08.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.09.version	TEXT	R/O	StringRange R/O
slot.09.model	TEXT	R/O	StringRange R/O
slot.09.name	TEXT		StringRange
slot.09.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.09.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.10.version	TEXT	R/O	StringRange R/O
slot.10.model	TEXT	R/O	StringRange R/O
slot.10.name	TEXT		StringRange
slot.10.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.10.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.11.version	TEXT	R/O	StringRange R/O
slot.11.model	TEXT	R/O	StringRange R/O
slot.11.name	TEXT		StringRange

name	*type*	*flags*	*range*
slot.11.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.11.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.12.version	TEXT	R/O	StringRange R/O
slot.12.model	TEXT	R/O	StringRange R/O
slot.12.name	TEXT		StringRange
slot.12.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.12.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.13.version	TEXT	R/O	StringRange R/O
slot.13.model	TEXT	R/O	StringRange R/O
slot.13.name	TEXT		StringRange
slot.13.card	CHOICE	R/O	EnumRange R/O (EMPTY TX-RX TX RX DTX DRX AMP SW FILTER OSC SPLTR SER TX-DWDM ETH DC-SWSPLTR EDGE-TX-DWDM TX-RX-LNB TX-LNB RX-LNB DTX-LNB DRX-LNB LNB-SAT6 DUAL-LNB SINGLE-LNB DWDM-TX-LNB)
slot.13.elapsedHours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.01.agc.1	CHOICE		EnumRange (OFF ON)
slot.01.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.01.sgc.1	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot.01.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.01.agc.2	CHOICE		EnumRange (OFF ON)
slot.01.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.01.sgc.2	CHOICE		EnumRange (OFF ON)
slot.01.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.agc.1	CHOICE		EnumRange (OFF ON)
slot.02.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.sgc.1	CHOICE		EnumRange (OFF ON)
slot.02.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.agc.2	CHOICE		EnumRange (OFF ON)
slot.02.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.02.sgc.2	CHOICE		EnumRange (OFF ON)
slot.02.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.03.agc.1	CHOICE		EnumRange (OFF ON)
slot.03.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.03.sgc.1	CHOICE		EnumRange (OFF ON)
slot.03.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.03.agc.2	CHOICE		EnumRange (OFF ON)
slot.03.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.03.sgc.2	CHOICE		EnumRange (OFF ON)
slot.03.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.04.agc.1	CHOICE		EnumRange (OFF ON)
slot.04.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.04.sgc.1	CHOICE		EnumRange (OFF ON)
slot.04.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.04.agc.2	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot.04.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.04.sgc.2	CHOICE		EnumRange (OFF ON)
slot.04.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.agc.1	CHOICE		EnumRange (OFF ON)
slot.05.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.sgc.1	CHOICE		EnumRange (OFF ON)
slot.05.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.agc.2	CHOICE		EnumRange (OFF ON)
slot.05.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.05.sgc.2	CHOICE		EnumRange (OFF ON)
slot.05.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.06.agc.1	CHOICE		EnumRange (OFF ON)
slot.06.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.06.sgc.1	CHOICE		EnumRange (OFF ON)
slot.06.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.06.agc.2	CHOICE		EnumRange (OFF ON)
slot.06.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.06.sgc.2	CHOICE		EnumRange (OFF ON)
slot.06.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.07.agc.1	CHOICE		EnumRange (OFF ON)
slot.07.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.07.sgc.1	CHOICE		EnumRange (OFF ON)
slot.07.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.07.agc.2	CHOICE		EnumRange (OFF ON)
slot.07.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.07.sgc.2	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
slot.07.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.agc.1	CHOICE		EnumRange (OFF ON)
slot.08.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.sgc.1	CHOICE		EnumRange (OFF ON)
slot.08.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.agc.2	CHOICE		EnumRange (OFF ON)
slot.08.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.08.sgc.2	CHOICE		EnumRange (OFF ON)
slot.08.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.09.agc.1	CHOICE		EnumRange (OFF ON)
slot.09.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.09.sgc.1	CHOICE		EnumRange (OFF ON)
slot.09.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.09.agc.2	CHOICE		EnumRange (OFF ON)
slot.09.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.09.sgc.2	CHOICE		EnumRange (OFF ON)
slot.09.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.10.agc.1	CHOICE		EnumRange (OFF ON)
slot.10.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.10.sgc.1	CHOICE		EnumRange (OFF ON)
slot.10.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.10.agc.2	CHOICE		EnumRange (OFF ON)
slot.10.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.10.sgc.2	CHOICE		EnumRange (OFF ON)
slot.10.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.agc.1	CHOICE		EnumRange (OFF ON)
slot.11.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.11.sgc.1	CHOICE		EnumRange (OFF ON)
slot.11.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.agc.2	CHOICE		EnumRange (OFF ON)
slot.11.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.11.sgc.2	CHOICE		EnumRange (OFF ON)
slot.11.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.12.agc.1	CHOICE		EnumRange (OFF ON)
slot.12.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.12.sgc.1	CHOICE		EnumRange (OFF ON)
slot.12.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.12.agc.2	CHOICE		EnumRange (OFF ON)
slot.12.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.12.sgc.2	CHOICE		EnumRange (OFF ON)
slot.12.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.13.agc.1	CHOICE		EnumRange (OFF ON)
slot.13.agcTarget.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.13.sgc.1	CHOICE		EnumRange (OFF ON)
slot.13.sgcGain.1	FLOAT		DoubleRange (0.00 .. 0.00)
slot.13.agc.2	CHOICE		EnumRange (OFF ON)
slot.13.agcTarget.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.13.sgc.2	CHOICE		EnumRange (OFF ON)
slot.13.sgcGain.2	FLOAT		DoubleRange (0.00 .. 0.00)
slot.01.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.01.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.02.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.03.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.03.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.04.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.05.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.05.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.06.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.07.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.07.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.08.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.09.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.09.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.10.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.11.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
slot.12.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.12.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfInpPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.optPower.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfGain.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.voltage.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.optCurrent.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfInpPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.optPower.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.rfGain.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.voltage.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.13.optCurrent.2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
slot.01.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.01.position	CHOICE		EnumRange (LEFT RIGHT)
slot.02.protection	CHOICE		EnumRange (AUTO MANUAL)

name	*type*	*flags*	*range*
slot.02.position	CHOICE		EnumRange (LEFT RIGHT)
slot.03.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.03.position	CHOICE		EnumRange (LEFT RIGHT)
slot.04.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.04.position	CHOICE		EnumRange (LEFT RIGHT)
slot.05.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.05.position	CHOICE		EnumRange (LEFT RIGHT)
slot.06.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.06.position	CHOICE		EnumRange (LEFT RIGHT)
slot.07.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.07.position	CHOICE		EnumRange (LEFT RIGHT)
slot.08.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.08.position	CHOICE		EnumRange (LEFT RIGHT)
slot.09.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.09.position	CHOICE		EnumRange (LEFT RIGHT)
slot.10.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.10.position	CHOICE		EnumRange (LEFT RIGHT)
slot.11.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.11.position	CHOICE		EnumRange (LEFT RIGHT)
slot.12.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.12.position	CHOICE		EnumRange (LEFT RIGHT)
slot.13.protection	CHOICE		EnumRange (AUTO MANUAL)
slot.13.position	CHOICE		EnumRange (LEFT RIGHT)
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Slot 1/1 Minor)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Slot 1/1 Major)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Slot 1/2 Minor)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Slot 1/2 Major)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Slot 2/1 Minor)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Slot 2/1 Major)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Slot 2/2 Minor)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Slot 2/2 Major)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Slot 3/1 Minor)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Slot 3/1 Major)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Slot 3/2 Minor)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Slot 3/2 Major)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Slot 4/1 Minor)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Slot 4/1 Major)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Slot 4/2 Minor)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Slot 4/2 Major)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Slot 5/1 Minor)

name	*type*	*flags*	*range*
faults.18	ALARM	R/O	AlarmFlagRange R/O (Slot 5/1 Major)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Slot 5/2 Minor)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Slot 5/2 Major)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Slot 6/1 Minor)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Slot 6/1 Major)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Slot 6/2 Minor)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Slot 6/2 Major)
faults.25	ALARM	R/O	AlarmFlagRange R/O (Slot 7/1 Minor)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Slot 7/1 Major)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Slot 7/2 Minor)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Slot 7/2 Major)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Slot 8/1 Minor)
faults.30	ALARM	R/O	AlarmFlagRange R/O (Slot 8/1 Major)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Slot 8/2 Minor)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Slot 8/2 Major)
faults.33	ALARM	R/O	AlarmFlagRange R/O (Slot 9/1 Minor)

name	*type*	*flags*	*range*
faults.34	ALARM	R/O	AlarmFlagRange R/O (Slot 9/1 Major)
faults.35	ALARM	R/O	AlarmFlagRange R/O (Slot 9/2 Minor)
faults.36	ALARM	R/O	AlarmFlagRange R/O (Slot 9/2 Major)
faults.37	ALARM	R/O	AlarmFlagRange R/O (Slot 10/1 Minor)
faults.38	ALARM	R/O	AlarmFlagRange R/O (Slot 10/1 Major)
faults.39	ALARM	R/O	AlarmFlagRange R/O (Slot 10/2 Minor)
faults.40	ALARM	R/O	AlarmFlagRange R/O (Slot 10/2 Major)
faults.41	ALARM	R/O	AlarmFlagRange R/O (Slot 11/1 Minor)
faults.42	ALARM	R/O	AlarmFlagRange R/O (Slot 11/1 Major)
faults.43	ALARM	R/O	AlarmFlagRange R/O (Slot 11/2 Minor)
faults.44	ALARM	R/O	AlarmFlagRange R/O (Slot 11/2 Major)
faults.45	ALARM	R/O	AlarmFlagRange R/O (Slot 12/1 Minor)
faults.46	ALARM	R/O	AlarmFlagRange R/O (Slot 12/1 Major)
faults.47	ALARM	R/O	AlarmFlagRange R/O (Slot 12/2 Minor)
faults.48	ALARM	R/O	AlarmFlagRange R/O (Slot 12/2 Major)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Slot 13/1 Minor)

name	*type*	*flags*	*range*
faults.50	ALARM	R/O	AlarmFlagRange R/O (Slot 13/1 Major)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Slot 13/2 Minor)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Slot 13/2 Major)
faults.53	ALARM	R/O	AlarmFlagRange R/O (PSU 1)
faults.54	ALARM	R/O	AlarmFlagRange R/O (PSU 2)
faults.55	ALARM	R/O	AlarmFlagRange R/O (System Low Voltage)
faults.56	ALARM	R/O	AlarmFlagRange R/O (System High Voltage)
faults.57	ALARM	R/O	AlarmFlagRange R/O (System Temperature)

9.721 ViaSat-4100-ACU

This page describes the device driver and the device window for the ViaSat 4100 antenna control unit.

The driver communicates with the ACU via TCP on port 6100.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first parameter page displays the antenna pointing and permits to set new pointing angles.
- --- The second parameter page displays / configures the parameters for the auto track and step track modes. It also lets you configure some parameters of the up to four tracking receivers used by the ACU.
- --- This page gives access to the target memories stored in the ACU for program track. You may manage the target memory list (add/delete/modify entries), start program tracking for a certain satellite or stop program tracking.
- --- This page gives access to the target memories stored in the ACU for adaptive track. You may manage the target memory list (add/delete entries), start adaptive tracking for a certain satellite or stop adaptive tracking.
- --- This page shows the state of the ACU's custom functions and lets you invoke these

by a button click.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the step track system.

parameter	***description***
slowJogStep	configures the slow jog steps commanded for the single arrow buttons for AZ/EL.
fastJogSpeed	configures the fast jog steps commanded for the double arrow buttons for AZ/EL and for the polarization axis.
azAxisMode	Sets the operation mode for this axis.
azAxisMode	Sets the operation mode for this axis.
azAxisMode	Sets the operation mode for this axis.
timeOfDay	Lets you set the internal clock of the ACU. The value is updated every 10 seconds, to change the time use exactly the same format as used for display.
pedestalPower	Switches the pedestal power ON/OFF

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.swRevision	TEXT	R/O	StringRange R/O
info.hwRevision	TEXT	R/O	StringRange R/O
info.fwRevision	TEXT	R/O	StringRange R/O
info.osRevision	TEXT	R/O	StringRange R/O
info.mcsRevision	TEXT	R/O	StringRange R/O
info.cpuRevision	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
info.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.antennaType	TEXT	R/O	StringRange R/O
info.noPol	INTEGER	R/O	IntegerRange R/O (0 .. 2)
info.noDI	INTEGER	R/O	IntegerRange R/O (0 .. 0)
i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
s	TEXT	R/O	StringRange R/O
internal.az	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.el	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ax3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ax4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.slowJogStep	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 10.00)
config.fastJogStep	FLOAT	SAVE SETUP	DoubleRange (0.01 .. 10.00)
config.azAxisMode	CHOICE	SETUP	EnumRange (UNDEFINED STANDBY MANUAL SLAVE RATE)
config.elAxisMode	CHOICE	SETUP	EnumRange (UNDEFINED STANDBY MANUAL SLAVE RATE)
config.plAxisMode	CHOICE	SETUP	EnumRange (UNDEFINED STANDBY MANUAL SLAVE RATE)
config.pl2AxisMode	CHOICE	SETUP	EnumRange (UNDEFINED STANDBY MANUAL SLAVE RATE)

name	*type*	*flags*	*range*
config.tiltAxisMode	CHOICE	SETUP	EnumRange (UNDEFINED STANDBY MANUAL SLAVE RATE)
config.timeOfDay	TEXT	SETUP	StringRange
config.pedestalPower	CHOICE	SETUP NOCOMPARE	EnumRange (OFF ON)
azimuth	FLOAT		DoubleRange (0.0000 .. 0.0000)
elevation	FLOAT		DoubleRange (0.0000 .. 0.0000)
polarization	FLOAT		DoubleRange (0.00 .. 0.00)
polarization2	FLOAT		DoubleRange (0.00 .. 0.00)
tilt	FLOAT		DoubleRange (0.00 .. 0.00)
frequency	FLOAT		DoubleRange DIS (0.000 .. 0.000)
attenuation	FLOAT		DoubleRange DIS (0.0 .. 50.0)
input	CHOICE		EnumRange DIS (1 2)
state.az	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.el	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.pl2	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.tilt	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.mode1	TEXT	R/O	StringRange R/O
state.mode2	TEXT	R/O	StringRange R/O
state.az.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.el.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.pl.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.pl2.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.tilt.moving	BOOLEAN	R/O	BooleanRange R/O (true false)
state.trackedSatellite	TEXT	R/O	StringRange R/O
state.frequency	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
jog	TEXT	NOPRESET	StringRange
motorStop	TEXT	NOPRESET	StringRange
atrk.enable	TEXT	NOPRESET	StringRange
atrk.disable	TEXT	NOPRESET	StringRange
atrk.diversity	CHOICE		EnumRange (OFF ON)
atrk.receiver	CHOICE		EnumRange (RCV1 RCV2 RCV3 RCV4)
atrk.az.state	TEXT	R/O	StringRange R/O
atrk.el.state	TEXT	R/O	StringRange R/O
atrk.rcv1.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
atrk.rcv2.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
atrk.rcv3.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
atrk.rcv4.threshold	FLOAT		DoubleRange (0.00 .. 0.00)
atrk.rcv1.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
atrk.rcv2.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
atrk.rcv3.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
atrk.rcv4.frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
atrk.rcv1.band	INTEGER		IntegerRange (0 .. 0)
atrk.rcv2.band	INTEGER		IntegerRange (0 .. 0)
atrk.rcv3.band	INTEGER		IntegerRange (0 .. 0)
atrk.rcv4.band	INTEGER		IntegerRange (0 .. 0)
atrk.rcv1.zero	TEXT	NOPRESET	StringRange
atrk.rcv2.zero	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
atrk.rcv3.zero	TEXT	NOPRESET	StringRange
atrk.rcv4.zero	TEXT	NOPRESET	StringRange
atrk.rcv1.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
atrk.rcv2.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
atrk.rcv3.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
atrk.rcv4.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
strk.enable	CHOICE		EnumRange (OFF ON)
strk.mode	TEXT	R/O	StringRange R/O
tgts.target	CHOICE		EnumRange (?)
tgts.add	TEXT	NOPRESET	StringRange
tgts.modify	TEXT	NOPRESET	StringRange
tgts.delete	TEXT	NOPRESET	StringRange
tgts.error	TEXT	R/O	StringRange R/O
mssn.point	TEXT	NOPRESET	StringRange
mssn.stop	TEXT	NOPRESET	StringRange
tgts.name	TEXT	NOPRESET	StringRange
tgts.tbase	CHOICE	NOPRESET	EnumRange (UTC GPS)
tgts.ecode	TEXT	NOPRESET	StringRange
tgts.epar.1	TEXT	NOPRESET	StringRange
tgts.epar.2	TEXT	NOPRESET	StringRange
tgts.epar.3	TEXT	NOPRESET	StringRange
tgts.epar.4	TEXT	NOPRESET	StringRange
tgts.epar.5	TEXT	NOPRESET	StringRange
tgts.epar.6	TEXT	NOPRESET	StringRange
tgts.epar.7	TEXT	NOPRESET	StringRange
tgts.epar.8	TEXT	NOPRESET	StringRange
tgts.epar.9	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
tgts.epar.10	TEXT	NOPRESET	StringRange
tgts.epar.11	TEXT	NOPRESET	StringRange
tgts.epar.12	TEXT	NOPRESET	StringRange
tgts.epar.13	TEXT	NOPRESET	StringRange
tgts.epar.14	TEXT	NOPRESET	StringRange
tgts.epar.15	TEXT	NOPRESET	StringRange
tgts.epar.16	TEXT	NOPRESET	StringRange
adpt.target	CHOICE		EnumRange (?)
adpt.name	TEXT	NOPRESET	StringRange
adpt.state	TEXT	R/O	StringRange R/O
adpt.add	TEXT	NOPRESET	StringRange
adpt.delete	TEXT	NOPRESET	StringRange
adpt.start	TEXT	NOPRESET	StringRange
adpt.stop	TEXT	NOPRESET	StringRange
custom.s11.name	TEXT	R/O	StringRange R/O
custom.s12.name	TEXT	R/O	StringRange R/O
custom.s13.name	TEXT	R/O	StringRange R/O
custom.s14.name	TEXT	R/O	StringRange R/O
custom.s15.name	TEXT	R/O	StringRange R/O
custom.s21.name	TEXT	R/O	StringRange R/O
custom.s22.name	TEXT	R/O	StringRange R/O
custom.s23.name	TEXT	R/O	StringRange R/O
custom.s24.name	TEXT	R/O	StringRange R/O
custom.s25.name	TEXT	R/O	StringRange R/O
custom.s31.name	TEXT	R/O	StringRange R/O
custom.s32.name	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
custom.s33.name	TEXT	R/O	StringRange R/O
custom.s34.name	TEXT	R/O	StringRange R/O
custom.s35.name	TEXT	R/O	StringRange R/O
custom.a7.name	TEXT	R/O	StringRange R/O
custom.a8.name	TEXT	R/O	StringRange R/O
custom.b7.name	TEXT	R/O	StringRange R/O
custom.b8.name	TEXT	R/O	StringRange R/O
custom.c7.name	TEXT	R/O	StringRange R/O
custom.c8.name	TEXT	R/O	StringRange R/O
custom.d7.name	TEXT	R/O	StringRange R/O
custom.d8.name	TEXT	R/O	StringRange R/O
custom.s11.state	TEXT	R/O	StringRange R/O
custom.s12.state	TEXT	R/O	StringRange R/O
custom.s13.state	TEXT	R/O	StringRange R/O
custom.s14.state	TEXT	R/O	StringRange R/O
custom.s15.state	TEXT	R/O	StringRange R/O
custom.s21.state	TEXT	R/O	StringRange R/O
custom.s22.state	TEXT	R/O	StringRange R/O
custom.s23.state	TEXT	R/O	StringRange R/O
custom.s24.state	TEXT	R/O	StringRange R/O
custom.s25.state	TEXT	R/O	StringRange R/O
custom.s31.state	TEXT	R/O	StringRange R/O
custom.s32.state	TEXT	R/O	StringRange R/O
custom.s33.state	TEXT	R/O	StringRange R/O
custom.s34.state	TEXT	R/O	StringRange R/O
custom.s35.state	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
custom.a7.state	TEXT	R/O	StringRange R/O
custom.a8.state	TEXT	R/O	StringRange R/O
custom.b7.state	TEXT	R/O	StringRange R/O
custom.b8.state	TEXT	R/O	StringRange R/O
custom.c7.state	TEXT	R/O	StringRange R/O
custom.c8.state	TEXT	R/O	StringRange R/O
custom.d7.state	TEXT	R/O	StringRange R/O
custom.d8.state	TEXT	R/O	StringRange R/O
custom.s11.on	TEXT	NOPRESET	StringRange
custom.s12.on	TEXT	NOPRESET	StringRange
custom.s13.on	TEXT	NOPRESET	StringRange
custom.s14.on	TEXT	NOPRESET	StringRange
custom.s15.on	TEXT	NOPRESET	StringRange
custom.s21.on	TEXT	NOPRESET	StringRange
custom.s22.on	TEXT	NOPRESET	StringRange
custom.s23.on	TEXT	NOPRESET	StringRange
custom.s24.on	TEXT	NOPRESET	StringRange
custom.s25.on	TEXT	NOPRESET	StringRange
custom.s31.on	TEXT	NOPRESET	StringRange
custom.s32.on	TEXT	NOPRESET	StringRange
custom.s33.on	TEXT	NOPRESET	StringRange
custom.s34.on	TEXT	NOPRESET	StringRange
custom.s35.on	TEXT	NOPRESET	StringRange
custom.a7.on	TEXT	NOPRESET	StringRange
custom.a8.on	TEXT	NOPRESET	StringRange
custom.b7.on	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
custom.b8.on	TEXT	NOPRESET	StringRange
custom.c7.on	TEXT	NOPRESET	StringRange
custom.c8.on	TEXT	NOPRESET	StringRange
custom.d7.on	TEXT	NOPRESET	StringRange
custom.d8.on	TEXT	NOPRESET	StringRange
custom.s11.off	TEXT	NOPRESET	StringRange
custom.s12.off	TEXT	NOPRESET	StringRange
custom.s13.off	TEXT	NOPRESET	StringRange
custom.s14.off	TEXT	NOPRESET	StringRange
custom.s15.off	TEXT	NOPRESET	StringRange
custom.s21.off	TEXT	NOPRESET	StringRange
custom.s22.off	TEXT	NOPRESET	StringRange
custom.s23.off	TEXT	NOPRESET	StringRange
custom.s24.off	TEXT	NOPRESET	StringRange
custom.s25.off	TEXT	NOPRESET	StringRange
custom.s31.off	TEXT	NOPRESET	StringRange
custom.s32.off	TEXT	NOPRESET	StringRange
custom.s33.off	TEXT	NOPRESET	StringRange
custom.s34.off	TEXT	NOPRESET	StringRange
custom.s35.off	TEXT	NOPRESET	StringRange
custom.a7.off	TEXT	NOPRESET	StringRange
custom.a8.off	TEXT	NOPRESET	StringRange
custom.b7.off	TEXT	NOPRESET	StringRange
custom.b8.off	TEXT	NOPRESET	StringRange
custom.c7.off	TEXT	NOPRESET	StringRange
custom.c8.off	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
custom.d7.off	TEXT	NOPRESET	StringRange
custom.d8.off	TEXT	NOPRESET	StringRange
di.num.1	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.2	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.3	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.4	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.5	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.6	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.7	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.8	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.9	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.10	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.11	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.12	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.13	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.14	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.15	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.16	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.17	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.18	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.19	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.20	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.21	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.22	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.23	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.24	INTEGER	R/O	IntegerRange R/O (0 .. 1)

name	*type*	*flags*	*range*
di.num.25	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.26	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.27	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.28	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.29	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.30	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.31	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.num.32	INTEGER	R/O	IntegerRange R/O (0 .. 1)
di.on.1	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.2	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.3	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.4	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.5	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.6	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.7	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.8	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.9	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.10	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.11	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.12	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.13	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.14	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.15	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.16	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.17	CHOICE	R/O	EnumRange R/O (OFF ON)

name	*type*	*flags*	*range*
di.on.18	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.19	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.20	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.21	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.22	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.23	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.24	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.25	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.26	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.27	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.28	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.29	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.30	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.31	CHOICE	R/O	EnumRange R/O (OFF ON)
di.on.32	CHOICE	R/O	EnumRange R/O (OFF ON)
gsa.target	TEXT		StringRange
gsa.dtrfreq	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.tle	TEXT		StringRange
gsa.startTrack	TEXT	NOPRESET	StringRange
gsa.stopTrack	TEXT	NOPRESET	StringRange
gsa.doPark	TEXT		StringRange
gsa.azPark	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.elPark	FLOAT		DoubleRange (0.000 .. 0.000)
gsa.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
faults.01	ALARM	R/O	AlarmFlagRange R/O (AZ Interlock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (EL Interlock)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (PL Interlock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (AZ Hi Limit)
faults.05	ALARM	R/O	AlarmFlagRange R/O (AZ Lo Limit)
faults.06	ALARM	R/O	AlarmFlagRange R/O (EL Hi Limit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EL Lo Limit)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PL Hi Limit)
faults.09	ALARM	R/O	AlarmFlagRange R/O (PL Lo Limit)
faults.10	ALARM	R/O	AlarmFlagRange R/O (MCS communication)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Diagnostic)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Local Lockout)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Emergency Stop)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Walkbox)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Non-Operational)
faults.16	ALARM	R/O	AlarmFlagRange R/O (System Warnings)
faults.17	ALARM	R/O	AlarmFlagRange R/O (System Faults)
faults.18	ALARM	R/O	AlarmFlagRange R/O (PL2 Interlock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PL2 Hi Limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (PL2 Lo Limit)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Tilt Interlock)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Tilt Hi Limit)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Tilt Lo Limit)

9.722 ViaSat-EBEM

This page describes the device driver and the device window for the ViaSat EBEM Satellite Modem. The driver requires the modem running EBEM version 2.03.02. Older version are known to be incompatible with this driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The IF parameters (frequency, level, etc.) and some global modem settings.
- --- The modulator settings, transmit clocking.
- --- The demodulator settings, receive clocking, doppler buffer.
- --- The Automatic Uplink Power Control settings.
- --- The data interface settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the DMD2050.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device. The default community setting of the ViaSat EBEM is "PUBLIC" (all uppercase letters)
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- password --- The ViaSat EBEM requires the M&C to log in with a password in order to get the permission to change any setting at the device. The password is set at the device's front panel (see paragraph "password" ant the end of this page).Enter the "SNMPv1 password" as set in the device here.

Read / Commit Usage

The ViaSat EBEM uses a different way of remote operation than other satellite modems do. Configuration parameters have to be read out by a 'readConfig' command, changes have to be applied explicitly by sending a 'commitConfig' command.

The sat-nms device driver implements these commands with the GUI buttons "READ SETTINGS" and "COMMIT CHANGES" which appear on each page of the device window where they apply. To update the displayed settings with values actually read from the device, click "READ SETTINGS". The button becomes yellow until the process has completed. To apply changes, click "COMMIT CHANGES". You may change several parameters, also on different pages and then click "COMMIT CHANGES". This sets all configuration parameters at the modem, commits the changes and reads back the actual settings. While this, the button is shown yellow.

This does not concern status variables like meter readings and fault flags. They are read by the driver on a regular base and updated automatically.

SNMPv1 Password

The modem requires to log in with a password before parameters can be changed. This password is defined by the crypto officer and set at the modem's front panel. The setup parameter "password" must be set to this password to enable parameter changes. With a wrong or outdated password, the modem will silently ignore any "COMMIT CHANGES" requests.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.ebemVersion	TEXT	R/O	StringRange R/O
info.serialNumber	TEXT	R/O	StringRange R/O
info.postStatus	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.bert.berStatus	TEXT	R/O	StringRange R/O
info.bert.patternSync	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (ON OFF)
config.password	TEXT	SAVE SETUP	StringRange
tx.modemType	CHOICE		EnumRange (MIL165A IESS308 IESS309 IESS310 EBEM CW)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.power	FLOAT		DoubleRange (-35.0 .. 10.0)
tx.on	CHOICE		EnumRange (OFF ON ON-INV)
tx.mod.type	CHOICE		EnumRange (BPSK QPSK OQPSK PSK8 APSK16)
tx.mod.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.ethDataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.fec	CHOICE		EnumRange (UNCODED 1/2-CEVD 3/4-CEVD 7/8-CEVD 1/2-CEVD-I 3/4-CEVD-I 7/8-CEVD-I 2/3-TCM 3/4-TCM 7/8-TCM 1/2-TURBO-1024 1/2-TURBO-4096 1/2-TURBO-16384 2/3-TURBO-1024 2/3-TURBO-4096 2/3-TURBO-16384 3/4-TURBO-1024 3/4-TURBO-4096 3/4-TURBO-16384 7/8-TURBO-1024 7/8-TURBO-4096 7/8-TURBO-16384 19/20-TURBO-1024 19/20-TURBO-4096)
tx.rs.encoding	CHOICE		EnumRange (OFF RS-126/112 RS-194/178 RS-208/192 RS-219/201 RS-225/205)
tx.mod.scrambling	CHOICE		EnumRange (OFF SYNC OM73SS IESSSS)

name	*type*	*flags*	*range*
tx.mod.differential	CHOICE		EnumRange (OFF ON)
tx.mod.spectrumInvert	CHOICE	R/O	EnumRange R/O (TX-OFF NORMAL INVERTED)
tx.aupc.enable	CHOICE		EnumRange (OFF ON)
tx.aupc.powerRange	FLOAT	R/O	DoubleRange R/O (0.0 .. 45.0)
tx.aupc.distEsNoRange	FLOAT	R/O	DoubleRange R/O (0.00 .. 2.00)
tx.aupc.status	TEXT	R/O	StringRange R/O
tx.aupc.powerSetPoint	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.aupc.actualDistEsNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.aupc.targetDistEsNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
rx.modemType	CHOICE		EnumRange (MIL165A IESS308 IESS309 IESS310 EBEM CW)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.dem.type	CHOICE		EnumRange (BPSK QPSK OQPSK PSK8 APSK16)
rx.dem.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
rx.dem.ethDataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
rx.dem.symbolRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
rx.dem.spectrumInvert	CHOICE		EnumRange (RX-OFF NORMAL INVERTED)

name	*type*	*flags*	*range*
rx.dem.fec	CHOICE		EnumRange (UNCODED 1/2-CEVD 3/4-CEVD 7/8-CEVD 1/2-CEVD-I 3/4-CEVD-I 7/8-CEVD-I 2/3-TCM 3/4-TCM 7/8-TCM 1/2-TURBO-1024 1/2-TURBO-4096 1/2-TURBO-16384 2/3-TURBO-1024 2/3-TURBO-4096 2/3-TURBO-16384 3/4-TURBO-1024 3/4-TURBO-4096 3/4-TURBO-16384 7/8-TURBO-1024 7/8-TURBO-4096 7/8-TURBO-16384 19/20-TURBO-1024 19/20-TURBO-4096)
rx.rs.encoding	CHOICE		EnumRange (OFF RS-126/112 RS-194/178 RS-208/192 RS-219/201 RS-225/205)
rx.dem.scrambling	CHOICE		EnumRange (OFF SYNC OM73SS IESSSS)
rx.dem.differential	CHOICE		EnumRange (OFF ON)
state.modemStatus	TEXT	R/O	StringRange R/O
state.encryption	TEXT	R/O	StringRange R/O
state.demodRole	TEXT	R/O	StringRange R/O
state.acqState	TEXT	R/O	StringRange R/O
rx.meter.ebno	FLOAT	R/O	DoubleRange R/O (0.00 .. 22.00)
rx.meter.esno	FLOAT	R/O	DoubleRange R/O (0.00 .. 25.00)
rx.meter.bufFill	FLOAT	R/O	DoubleRange R/O (0 .. 100)
rx.meter.signal	FLOAT	R/O	DoubleRange R/O (-100 .. 0)
rx.meter.level	FLOAT	R/O	DoubleRange R/O (-100 .. 0)
rx.meter.frequencyOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
rx.meter.ber	TEXT	R/O	StringRange R/O
config.runBit	CHOICE	SETUP	EnumRange (none Non-destructive Destructive 1 Destructive 2 Destructive 3)
readConfig	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
commitConfig	TEXT	NOPRESET	StringRange
internal.cfs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.i	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RAM)
faults.02	ALARM	R/O	AlarmFlagRange R/O (FLASH)
faults.03	ALARM	R/O	AlarmFlagRange R/O (FPGA)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Power Supply)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Encryption)
faults.07	ALARM	R/O	AlarmFlagRange R/O (POST)
faults.08	ALARM	R/O	AlarmFlagRange R/O (BBI)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Bit Sync)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Carrier Lock)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Emb.Ch. Sync)

9.723 VioIP-End-Point

The VioIP-End-Point device is used by the sat-nms VioIP software to monitor and control one single connection end point. The device translates the abstract commands for starting or stopping a connection it receives from the VioIP server into device specific parameter settings. For monitoring it checks certain alarms of the monitored device and filters them in a way that connection specific faults are masked while the device actually does not service a connection.

Connection States

The device knows a state "idle" which means there is actually no connection running, the device is ready to receive a start command from the VioIP server. When the device receives a connection start command, it first applies the device preset defined in the connection profile of the upcoming connection. After a short time the device then sends the connection specific commands as defined with the start macro at the second device screen page. When these commands are sent, the device waits "startDelay" seconds before it changes to the state "running" and now unmasks the connection specific devices faults.

The device now remains in the "running" state until a stop command is received from the

server. Then the device masks the connection specific faults again, reads the packet counters from the device and finally applies the stop commands to the device. It stays for "stopDelay" seconds in this state and now sends the counter reset commands to the device. The device returns back to "idle" state.

Start / Stop Commands

The commands sent to the device in order to start or stop a connection are defined at the second page of the device window. The format is much like the macro language used by the sat-nms software, commands typically consist of the destination message ID, a space character and the value to be set. Empty lines and lines starting with the comment character '#' are ignored.

There are four commands the device uses during the life cycle of a connection:

- **start unicast connection commands** This macro is executed to start a unicast connection.
- **start multicast connection commands** This macro is executed to start a multicast connection.
- **stop connection commands** This macro is executed to stop a connection. This macro also should contain the commands to read the packet counters from the device.
- **reset statistics commands** This macro is executed to reset the packet counters in the device. For device types which automatically clear the counters when a new connection starts, this may be empty.

Command Parameters

The macro language used for these commands knows a number of placeholders which may be used to insert parameters which are not known before the connection starts. These are:

- **%DEV** --- The name of the physical device we are controlling (the name is defined as a configuration parameter of this device).
- **%SRC** --- The source IP address of the stream in case of a unicast connection. The value of this parameter is taken from the server's start command.
- **%DST** --- The destination IP address of the stream in case of a unicast connection. The value of this parameter is taken from the server's start command.
- **%MCT** --- The multicast group / IP address of the stream in case of a unicast connection. The value of this parameter is taken from the server's start command.
- **%PNR** --- The port number to be used. The value of this parameter is taken from the server's start command as well.

Beside these parameters, the macros knew three commands to read the packet counters from the physical device. For these commands the sequence of words is just the other way round: The line starts with one of the commands followed by a space character and the message ID of the parameter to read. The latter may contain the %DEV placeholder for the name of the physical device. The commands to read the packet counters are:

- **%TPC** --- Read the total number of packets received.
- **%CPC** --- Read the number of corrected packets received.

- **%UPC** --- Read the number of uncorrected packets received.

The macros also may contain commands directed to other devices than the %DEV device. In a start macro you may command a switch or matrix port in order to select the proper input signal for an encoder for example.

Finally, when you create a VioIP-End-Point device using the "Add New Device" function in the VioIP Network Configuration Application , you may activate the checkmark "initialize end point device with library settings". In this case the VioIP Network Configuration Application fills the start / stop macros with defaults suitable for the physical device you selected.

Examples

Below you find a number of examples for command macros. The shall illustrate the usage of the command language and the placeholders described above:

Start a unicast connection from a Ateme-Kyrion-CM5000 Encoder

Start a multicast connection from a Ateme-Kyrion-CM5000 Encoder

Stop a connection at a Ateme-DR5000 IRD, read the packet counters before actually stopping the connections

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- This page shows the actual state of the device together with the connection parameter the VioIP server has sent with the recent start command
- --- This page lets you configure the commands sent to the physical device to start or stop a connection
- --- The faults and fault mask page.
- --- The device info page.
- --- The maintenance page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***

parameter	*description*
simulation	With the simulation mode you may use the VioIP-End-Point without a physical device attached. This is for testing purposes only, you should leave this parameter at OFF
moduleIndex	If the physical device controlled by this end point is single encoder or decoder, leave this parameter at 0. With devices which provide multiple instances of encoders or decoders, enter a number 1..n which addresses the module within the device.
device	The name of the physical device
fileRecorder	The name of a file recorder used to log the quality data of connections processed by this end point.
deviceFault1Id deviceFault2Id deviceFault3Id	You may enter up to three parameter IDs of fault flags which shall be monitored only if a connection is actually running. Fault conditions like "input signal missing" or "stream lock missing" are typical faults you may want to monitor this way.
startDelay	The device starts to monitor the faults above this number of seconds after it receives the start command this delay gives the device a chance to "settle" and to clear its faults after it has been set for the upcoming connection
stopDelay	The device clears the device's packet counters this number of seconds after it has received the stop command from the server

When creating a VioIP-End-Point device using the "Add New Device" function in the[VioIP Network Configuration Application] you may activate the checkmark "initialize end point device with library settings". In this case the VioIP Network Configuration Application sets all the configration parameters above to defaults suitable for the physical device you selected.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.simulation	CHOICE	SAVE SETUP	EnumRange (OFF ON)

name	*type*	*flags*	*range*
config.moduleIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 16)
config.device	TEXT	SAVE SETUP	StringRange
config.fileRecorder	TEXT	SAVE SETUP	StringRange
config.deviceFault1Id	TEXT	SAVE SETUP	StringRange
config.deviceFault2Id	TEXT	SAVE SETUP	StringRange
config.deviceFault3Id	TEXT	SAVE SETUP	StringRange
config.startDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
config.stopDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 30)
cmd	TEXT	NOPRESET	StringRange
cmdmem	TEXT	SAVE NOPRESET	StringRange
startUnicastCmds	TEXT	SAVE	StringRange
startMulticastCmds	TEXT	SAVE	StringRange
stopCmds	TEXT	SAVE	StringRange
counterResetCmds	TEXT	SAVE	StringRange
mcast	TEXT	R/O	StringRange R/O
srcip	TEXT	R/O	StringRange R/O
dstip	TEXT	R/O	StringRange R/O
running	BOOLEAN	R/O	BooleanRange R/O (true false)
preset	TEXT	R/O	StringRange R/O
side	TEXT	R/O	StringRange R/O
port	INTEGER	R/O	IntegerRange R/O (0 .. 0)
totalPackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
correctedPackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
uncorrectedPackets	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Device Fault 1)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Device Fault 2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Device Fault 3)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Device Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Device Communication)

9.724 Viprinet-HUB-SNMP

This page describes the Viprinet-HUB-SNMP device driver and the device window for this driver.

The Viprinet-Tunnel-SNMP driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-Tunnel-SNMP monitors the status and performance of one VPN tunnel in a Viprinet Hub and not the Hub itself. There is one instance of this device to configured for each VPN tunnel to monitor. Control of parameters is not possible.

Viprinet-Rugged-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the general state of the router's WAN interfaces. Control of parameters is not possible.

Viprinet-Rugged-Router-HTTPS monitors detailed status parameters of a Viprinet Router device, including the detailed state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface and also control of WiFi access point, Roaming and APN.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The status parameters of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The Viprinet hub / router devices use 'public' as read community by default.

Please note that multiple device driver instances which are talking to the same device must be configured with the same interface. Also the Viprinet-Tunnel-SNMP devices must be configured within the same interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.model	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.mode	TEXT	R/O	StringRange R/O
info.modeMsg	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.firmwareStatus	TEXT	R/O	StringRange R/O
config.systemTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.cpuTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
averageCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
memoryUsage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
systemTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cpuTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.1.enabled	TEXT	R/O	StringRange R/O
fan.2.enabled	TEXT	R/O	StringRange R/O
fan.1.status	TEXT	R/O	StringRange R/O
fan.2.status	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (System temp)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CPU temp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 2)

9.725 Viprinet-HUB

This page describes the Viprinet-HUB device driver and the device window for this driver.

The Viprinet-HUB driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB monitors some general status parameters of a Viprinet Hub device.

Viprinet-Router monitors some general status parameters of a Viprinet Router device, including the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface.

Viprinet-Tunnel monitors the status and performance of one VPN tunnel in a Viprinet Hub. There is one instance of this device to configured for each VPN tunnel to monitor.

All three device drivers use the ssh command line interface of the Viprinet device for monitoring and control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The status parameters of the device
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The ssh address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp username password*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts ssh connections. The port number may be omitted, the driver uses the standard port 22 in this case.
  - *username* --- The user name to log in at the device. The user name is mandatory.
  - *password* --- The password to login at the device. This is mandatory as well.

The Viprinet hub / router devices use 'root' as username and 'viprinet' as password by default.

Please note that multiple device driver instances which are talking to the same device must be configured with the same interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
cpuTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
systemTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.1.enabled	TEXT	R/O	StringRange R/O
fan.2.enabled	TEXT	R/O	StringRange R/O
fan.1.status	TEXT	R/O	StringRange R/O
fan.2.status	TEXT	R/O	StringRange R/O
memoryUsage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
currentCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
averageCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
quietMode	CHOICE		EnumRange (YES NO)

9.726 Viprinet-Hub-Redundancy-HTTPS

This page describes the Viprinet-Hub-Redundancy-HTTPS device driver and the device window for this driver.

The Viprinet-Hub-Redundancy-HTTPS device is used in a Viprinet hub hotspare configuration consisting of main and backup hub. After the backup hub has replaced the main hub, a manual release via the webpage is necessary. This driver is used to release the temporary replacement Hub from replacing the main hub via the sat-nms M&C System.

The Viprinet-Hub-Redundancy-HTTPS device driver is part of a group of device drivers used to

monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB-SNMP monitors some general status parameters of a Viprinet Hub device like temperature and cpu load - not VPN tunnel details. Control of parameters is not possible.

Viprinet-Tunnel-SNMP monitors the status and performance of one VPN tunnel in a Viprinet Hub and not the Hub itself. There is one instance of this device to configured for each VPN tunnel to monitor. Control of parameters is not possible.

Viprinet-Rugged-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the general state of the router's WAN interfaces. Control of parameters is not possible.

Viprinet-Rugged-Router-HTTPS monitors detailed status parameters of a Viprinet Router device, including the detailed state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface and also control of WiFi access point, Roaming and APN.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The control page.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	The IP address of the device to control in 'dotted quad' notation. The address may be followed by a port number, separated by a colon. If no port number is specified, the HTTPS standard port (443) is used.
userName	The user name to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'root'.
password	The password to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'viprinet'.

During normal operation the driver is not logged in at the Viprinet Hub. Once the button "release hub" is pressed, the device driver performs a login and trigger the release procedure. Both Viprinet Hubs perform a reboot and go back to normal operation state as main hub and standby hub.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sessionId	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
info.login	TEXT	R/O	StringRange R/O
config.userName	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
release	TEXT	NOPRESET	StringRange

9.727 Viprinet-Router-HTTPS

This page describes the Viprinet-Router-HTTPS device driver and the device window for this driver.

The Viprinet-Router-HTTPS driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB monitors some general status parameters of a Viprinet Hub device.

Viprinet-Router monitors some general status parameters of a Viprinet Router device, including

the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface.

Viprinet-Router-HTTPS monitors some general status parameters of a Viprinet Router device, including the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface. This driver uses the HTTPS protocol to interrogate / control the router device rather than SSH

Viprinet-Tunnel monitors the status and performance of one VPN tunnel in a Viprinet Hub. There is one instance of this device to configured for each VPN tunnel to monitor.

All three device drivers use the ssh command line interface of the Viprinet device for monitoring and control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The general status parameters of the device
- --- The status parameter of the interface slots / WAN interfaces of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	The IP address of the device to control in 'dotted quad' notation. The address may be followed by a port number, separated by a colon. If no port number is specified, the HTTPS standard port (443) is used.
userName	The user name to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'root'.
password	The password to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'viprinet'.
backplaneSlots	The number of backplane slots this router has. Must be set to the physical number of slots available with this router model. Defaults to 6.

REMARKS

The driver polls the device at a very low cycle rate (about once every two minutes). This is

because answering the status requests from the M&C requires much CPU power in the router device. You can use the buttons marked with in order to update the status display for a certain WAN interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sessionId	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
config.userName	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
config.hasAP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.backplaneSlots	INTEGER	SAVE SETUP	IntegerRange (1 .. 6)
reset	TEXT	NOPRESET	StringRange
slot.1.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.2.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.3.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.4.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.5.configured	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.6.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.1.name	TEXT	R/O	StringRange R/O
slot.1.status	TEXT	R/O	StringRange R/O
slot.1.reconnect	TEXT	NOPRESET	StringRange
slot.1.reset	TEXT	NOPRESET	StringRange
slot.1.update	TEXT	NOPRESET	StringRange
slot.1.moduleInfo	TEXT	R/O	StringRange R/O
slot.1.network	TEXT	R/O	StringRange R/O
slot.1.serviceType	TEXT	R/O	StringRange R/O
slot.1.roaming	CHOICE		EnumRange (NO YES)
slot.1.enabled	CHOICE		EnumRange (NO YES)
slot.1.apn	TEXT		StringRange
slot.1.autoApn	CHOICE		EnumRange (NO YES)
slot.2.name	TEXT	R/O	StringRange R/O
slot.2.status	TEXT	R/O	StringRange R/O
slot.2.reconnect	TEXT	NOPRESET	StringRange
slot.2.reset	TEXT	NOPRESET	StringRange
slot.2.update	TEXT	NOPRESET	StringRange
slot.2.moduleInfo	TEXT	R/O	StringRange R/O
slot.2.network	TEXT	R/O	StringRange R/O
slot.2.serviceType	TEXT	R/O	StringRange R/O
slot.2.roaming	CHOICE		EnumRange (NO YES)
slot.2.enabled	CHOICE		EnumRange (NO YES)
slot.2.apn	TEXT		StringRange
slot.2.autoApn	CHOICE		EnumRange (NO YES)
slot.3.name	TEXT	R/O	StringRange R/O
slot.3.status	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot.3.reconnect	TEXT	NOPRESET	StringRange
slot.3.reset	TEXT	NOPRESET	StringRange
slot.3.update	TEXT	NOPRESET	StringRange
slot.3.moduleInfo	TEXT	R/O	StringRange R/O
slot.3.network	TEXT	R/O	StringRange R/O
slot.3.serviceType	TEXT	R/O	StringRange R/O
slot.3.roaming	CHOICE		EnumRange (NO YES)
slot.3.enabled	CHOICE		EnumRange (NO YES)
slot.3.apn	TEXT		StringRange
slot.3.autoApn	CHOICE		EnumRange (NO YES)
slot.4.name	TEXT	R/O	StringRange R/O
slot.4.status	TEXT	R/O	StringRange R/O
slot.4.reconnect	TEXT	NOPRESET	StringRange
slot.4.reset	TEXT	NOPRESET	StringRange
slot.4.update	TEXT	NOPRESET	StringRange
slot.4.moduleInfo	TEXT	R/O	StringRange R/O
slot.4.network	TEXT	R/O	StringRange R/O
slot.4.serviceType	TEXT	R/O	StringRange R/O
slot.4.roaming	CHOICE		EnumRange (NO YES)
slot.4.enabled	CHOICE		EnumRange (NO YES)
slot.4.apn	TEXT		StringRange
slot.4.autoApn	CHOICE		EnumRange (NO YES)
slot.5.name	TEXT	R/O	StringRange R/O
slot.5.status	TEXT	R/O	StringRange R/O
slot.5.reconnect	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
slot.5.reset	TEXT	NOPRESET	StringRange
slot.5.update	TEXT	NOPRESET	StringRange
slot.5.moduleInfo	TEXT	R/O	StringRange R/O
slot.5.network	TEXT	R/O	StringRange R/O
slot.5.serviceType	TEXT	R/O	StringRange R/O
slot.5.roaming	CHOICE		EnumRange (NO YES)
slot.5.enabled	CHOICE		EnumRange (NO YES)
slot.5.apn	TEXT		StringRange
slot.5.autoApn	CHOICE		EnumRange (NO YES)
slot.6.name	TEXT	R/O	StringRange R/O
slot.6.status	TEXT	R/O	StringRange R/O
slot.6.reconnect	TEXT	NOPRESET	StringRange
slot.6.reset	TEXT	NOPRESET	StringRange
slot.6.update	TEXT	NOPRESET	StringRange
slot.6.moduleInfo	TEXT	R/O	StringRange R/O
slot.6.network	TEXT	R/O	StringRange R/O
slot.6.serviceType	TEXT	R/O	StringRange R/O
slot.6.roaming	CHOICE		EnumRange (NO YES)
slot.6.enabled	CHOICE		EnumRange (NO YES)
slot.6.apn	TEXT		StringRange
slot.6.autoApn	CHOICE		EnumRange (NO YES)
ap.enabled	CHOICE		EnumRange DIS (NO YES)
i	TEXT	R/O	StringRange R/O
m	TEXT	R/O	StringRange R/O
p	TEXT	R/O	StringRange R/O
n	TEXT	R/O	StringRange R/O
c	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
s	TEXT	R/O	StringRange R/O
nw	TEXT	R/O	StringRange R/O
st	TEXT	R/O	StringRange R/O
mobile	TEXT	R/O	StringRange R/O

9.728 Viprinet-Router-SNMP

This page describes the Viprinet-Router device driver and the device window for this driver.

The Viprinet-Router-SNMP driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB-SNMP monitors some general status parameters of a Viprinet Hub device.

Viprinet-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface.

Viprinet-Tunnel-SNMP monitors the status and performance of one VPN tunnel in a Viprinet Hub and not the Hub itself. There is one instance of this device to configured for each VPN tunnel to monitor. Control of parameters is not possible.

Viprinet-Rugged-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the general state of the router's WAN interfaces. Control of parameters is not possible.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The general status parameters of the device
- --- The status parameter of the interface slots / WAN interfaces of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The ssh address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp username password*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts ssh connections. The port number may be omitted, the driver uses the standard port 22 in this case.
  - *username* --- The user name to log in at the device. The user name is mandatory.
  - *password* --- The password to login at the device. This is mandatory as well.

The Viprinet hub / router devices use 'root' as username and 'viprinet' as password by default.

Please note that multiple device driver instances which are talking to the same device must be configured with the same interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.mode	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.modeMsg	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.firmwareStatus	TEXT	R/O	StringRange R/O
config.systemTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.cpuTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
averageCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
memoryUsage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
systemTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cpuTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.1.enabled	TEXT	R/O	StringRange R/O
fan.2.enabled	TEXT	R/O	StringRange R/O
fan.1.status	TEXT	R/O	StringRange R/O
fan.2.status	TEXT	R/O	StringRange R/O
backplaneSlots	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.1.name	TEXT	R/O	StringRange R/O DIS
slot.1.enabled	TEXT	R/O	StringRange R/O DIS
slot.1.status	TEXT	R/O	StringRange R/O DIS
slot.2.name	TEXT	R/O	StringRange R/O DIS
slot.2.enabled	TEXT	R/O	StringRange R/O DIS
slot.2.status	TEXT	R/O	StringRange R/O DIS
slot.3.name	TEXT	R/O	StringRange R/O DIS
slot.3.enabled	TEXT	R/O	StringRange R/O DIS
slot.3.status	TEXT	R/O	StringRange R/O DIS
slot.4.name	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
slot.4.enabled	TEXT	R/O	StringRange R/O DIS
slot.4.status	TEXT	R/O	StringRange R/O DIS
slot.5.name	TEXT	R/O	StringRange R/O DIS
slot.5.enabled	TEXT	R/O	StringRange R/O DIS
slot.5.status	TEXT	R/O	StringRange R/O DIS
slot.6.name	TEXT	R/O	StringRange R/O DIS
slot.6.enabled	TEXT	R/O	StringRange R/O DIS
slot.6.status	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (System temp)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CPU temp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 2)

9.729 Viprinet-Router

This page describes the Viprinet-Router device driver and the device window for this driver.

The Viprinet-Router driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB monitors some general status parameters of a Viprinet Hub device.

Viprinet-Router monitors some general status parameters of a Viprinet Router device, including the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface.

Viprinet-Tunnel monitors the status and performance of one VPN tunnel in a Viprinet Hub. There is one instance of this device to configured for each VPN tunnel to monitor.

All three device drivers use the ssh command line interface of the Viprinet device for monitoring and control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The general status parameters of the device
- --- The status parameter of the interface slots / WAN interfaces of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The ssh address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp username password*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts ssh connections. The port number may be omitted, the driver uses the standard port 22 in this case.
  - *username* --- The user name to log in at the device. The user name is mandatory.
  - *password* --- The password to login at the device. This is mandatory as well.

The Viprinet hub / router devices use 'root' as username and 'viprinet' as password by default.

Please note that multiple device driver instances which are talking to the same device must be configured with the same interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
backplaneSlots	INTEGER	R/O	IntegerRange R/O (0 .. 0)
memoryUsage	INTEGER	R/O	IntegerRange R/O (0 .. 0)
currentCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
averageCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.1.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.2.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.3.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.4.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.5.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.6.available	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.1.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.2.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.3.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.4.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.5.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.6.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.1.reconnect	TEXT	NOPRESET	StringRange
slot.1.reset	TEXT	NOPRESET	StringRange
slot.1.name	TEXT	R/O	StringRange R/O
slot.1.status	TEXT	R/O	StringRange R/O
slot.2.reconnect	TEXT	NOPRESET	StringRange
slot.2.reset	TEXT	NOPRESET	StringRange
slot.2.name	TEXT	R/O	StringRange R/O
slot.2.status	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot.3.reconnect	TEXT	NOPRESET	StringRange
slot.3.reset	TEXT	NOPRESET	StringRange
slot.3.name	TEXT	R/O	StringRange R/O
slot.3.status	TEXT	R/O	StringRange R/O
slot.4.reconnect	TEXT	NOPRESET	StringRange
slot.4.reset	TEXT	NOPRESET	StringRange
slot.4.name	TEXT	R/O	StringRange R/O
slot.4.status	TEXT	R/O	StringRange R/O
slot.5.reconnect	TEXT	NOPRESET	StringRange
slot.5.reset	TEXT	NOPRESET	StringRange
slot.5.name	TEXT	R/O	StringRange R/O
slot.5.status	TEXT	R/O	StringRange R/O
slot.6.reconnect	TEXT	NOPRESET	StringRange
slot.6.reset	TEXT	NOPRESET	StringRange
slot.6.name	TEXT	R/O	StringRange R/O
slot.6.status	TEXT	R/O	StringRange R/O
internal.name	TEXT	R/O	StringRange R/O
internal.string	TEXT	R/O	StringRange R/O

9.730 Viprinet-Rugged-Router-HTTPS

This page describes the Viprinet-Rugged-Router-HTTPS device driver and the device window for this driver.

The Viprinet-Rugged-Router-HTTPS driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB-SNMP monitors some general status parameters of a Viprinet Hub device like temperature and cpu load - not VPN tunnel details. Control of parameters is not possible.

Viprinet-Tunnel-SNMP monitors the status and performance of one VPN tunnel in a Viprinet Hub and not the Hub itself. There is one instance of this device to configured for each VPN tunnel to

monitor. Control of parameters is not possible.

Viprinet-Rugged-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the general state of the router's WAN interfaces. Control of parameters is not possible.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The general status parameters of the device
- --- The status parameter of the interface slots / WAN interfaces of the device.
- --- The status parameters of each channel
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	The IP address of the device to control in 'dotted quad' notation. The address may be followed by a port number, separated by a colon. If no port number is specified, the HTTPS standard port (443) is used.
userName	The user name to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'root'.
password	The password to log in at the Viprinet Web-GUI. The default value used by Viprinet routers out of the box is 'viprinet'.
hasAP	Switch on/off the WiFi AP control function of the device driver.
backplaneSlots	The number of backplane slots this router has. Must be set to the physical number of slots available with this router model. Defaults to 6.

REMARKS

The driver polls the device at a very low cycle rate (about once every two minutes). This is because answering the status requests from the M&C requires much CPU power in the router device. You can use the buttons marked with in order to update the status display for a

certain WAN interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.sessionId	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
info.login	TEXT	R/O	StringRange R/O
info.lcycles	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.userName	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
config.hasAP	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.backplaneSlots	INTEGER	SAVE SETUP	IntegerRange (1 .. 6)
reset	TEXT	NOPRESET	StringRange
slot.1.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.2.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.3.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.4.configured	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
slot.5.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.6.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.1.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.2.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.3.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.4.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.5.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.6.hasPD	BOOLEAN	R/O	BooleanRange R/O (true false)
slot.1.name	TEXT	R/O	StringRange R/O
slot.1.status	TEXT	R/O	StringRange R/O
slot.1.reconnect	TEXT	NOPRESET	StringRange
slot.1.reset	TEXT	NOPRESET	StringRange
slot.1.update	TEXT	NOPRESET	StringRange
slot.1.moduleInfo	TEXT	R/O	StringRange R/O
slot.1.network	TEXT	R/O	StringRange R/O
slot.1.serviceType	TEXT	R/O	StringRange R/O
slot.1.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.1.bandInfo	TEXT	R/O	StringRange R/O
slot.1.roamStat	TEXT	R/O	StringRange R/O
slot.1.roaming	CHOICE		EnumRange (NO YES)
slot.1.enabled	CHOICE		EnumRange (NO YES)
slot.1.apn	TEXT		StringRange
slot.1.autoApn	CHOICE		EnumRange (NO YES)
slot.2.name	TEXT	R/O	StringRange R/O
slot.2.status	TEXT	R/O	StringRange R/O
slot.2.reconnect	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
slot.2.reset	TEXT	NOPRESET	StringRange
slot.2.update	TEXT	NOPRESET	StringRange
slot.2.moduleInfo	TEXT	R/O	StringRange R/O
slot.2.network	TEXT	R/O	StringRange R/O
slot.2.serviceType	TEXT	R/O	StringRange R/O
slot.2.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.2.bandInfo	TEXT	R/O	StringRange R/O
slot.2.roamStat	TEXT	R/O	StringRange R/O
slot.2.roaming	CHOICE		EnumRange (NO YES)
slot.2.enabled	CHOICE		EnumRange (NO YES)
slot.2.apn	TEXT		StringRange
slot.2.autoApn	CHOICE		EnumRange (NO YES)
slot.3.name	TEXT	R/O	StringRange R/O
slot.3.status	TEXT	R/O	StringRange R/O
slot.3.reconnect	TEXT	NOPRESET	StringRange
slot.3.reset	TEXT	NOPRESET	StringRange
slot.3.update	TEXT	NOPRESET	StringRange
slot.3.moduleInfo	TEXT	R/O	StringRange R/O
slot.3.network	TEXT	R/O	StringRange R/O
slot.3.serviceType	TEXT	R/O	StringRange R/O
slot.3.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.3.bandInfo	TEXT	R/O	StringRange R/O
slot.3.roamStat	TEXT	R/O	StringRange R/O
slot.3.roaming	CHOICE		EnumRange (NO YES)
slot.3.enabled	CHOICE		EnumRange (NO YES)
slot.3.apn	TEXT		StringRange
slot.3.autoApn	CHOICE		EnumRange (NO YES)

name	*type*	*flags*	*range*
slot.4.name	TEXT	R/O	StringRange R/O
slot.4.status	TEXT	R/O	StringRange R/O
slot.4.reconnect	TEXT	NOPRESET	StringRange
slot.4.reset	TEXT	NOPRESET	StringRange
slot.4.update	TEXT	NOPRESET	StringRange
slot.4.moduleInfo	TEXT	R/O	StringRange R/O
slot.4.network	TEXT	R/O	StringRange R/O
slot.4.serviceType	TEXT	R/O	StringRange R/O
slot.4.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.4.bandInfo	TEXT	R/O	StringRange R/O
slot.4.roamStat	TEXT	R/O	StringRange R/O
slot.4.roaming	CHOICE		EnumRange (NO YES)
slot.4.enabled	CHOICE		EnumRange (NO YES)
slot.4.apn	TEXT		StringRange
slot.4.autoApn	CHOICE		EnumRange (NO YES)
slot.5.name	TEXT	R/O	StringRange R/O
slot.5.status	TEXT	R/O	StringRange R/O
slot.5.reconnect	TEXT	NOPRESET	StringRange
slot.5.reset	TEXT	NOPRESET	StringRange
slot.5.update	TEXT	NOPRESET	StringRange
slot.5.moduleInfo	TEXT	R/O	StringRange R/O
slot.5.network	TEXT	R/O	StringRange R/O
slot.5.serviceType	TEXT	R/O	StringRange R/O
slot.5.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.5.bandInfo	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
slot.5.roamStat	TEXT	R/O	StringRange R/O
slot.5.roaming	CHOICE		EnumRange (NO YES)
slot.5.enabled	CHOICE		EnumRange (NO YES)
slot.5.apn	TEXT		StringRange
slot.5.autoApn	CHOICE		EnumRange (NO YES)
slot.6.name	TEXT	R/O	StringRange R/O
slot.6.status	TEXT	R/O	StringRange R/O
slot.6.reconnect	TEXT	NOPRESET	StringRange
slot.6.reset	TEXT	NOPRESET	StringRange
slot.6.update	TEXT	NOPRESET	StringRange
slot.6.moduleInfo	TEXT	R/O	StringRange R/O
slot.6.network	TEXT	R/O	StringRange R/O
slot.6.serviceType	TEXT	R/O	StringRange R/O
slot.6.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.6.bandInfo	TEXT	R/O	StringRange R/O
slot.6.roamStat	TEXT	R/O	StringRange R/O
slot.6.roaming	CHOICE		EnumRange (NO YES)
slot.6.enabled	CHOICE		EnumRange (NO YES)
slot.6.apn	TEXT		StringRange
slot.6.autoApn	CHOICE		EnumRange (NO YES)
ap.enabled	CHOICE		EnumRange DIS (NO YES)
i	TEXT	R/O	StringRange R/O
m	TEXT	R/O	StringRange R/O
p	TEXT	R/O	StringRange R/O
n	TEXT	R/O	StringRange R/O
c	TEXT	R/O	StringRange R/O
s	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
nw	TEXT	R/O	StringRange R/O
st	TEXT	R/O	StringRange R/O
mobile	TEXT	R/O	StringRange R/O
pd.signal	INTEGER	R/O	IntegerRange R/O (0 .. 0)
pd.serviceType	TEXT	R/O	StringRange R/O
pd.status	TEXT	R/O	StringRange R/O
pd.roamStat	TEXT	R/O	StringRange R/O
pd.network	TEXT	R/O	StringRange R/O
pd.bandInfo	TEXT	R/O	StringRange R/O
apn	TEXT	R/O	StringRange R/O

9.731 Viprinet-Rugged-Router-SNMP

This page describes the Viprinet-Rugged-Router-SNMP device driver and the device window for this driver.

The Viprinet-Rugged-Router-SNMP driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB-SNMP monitors some general status parameters of a Viprinet Hub device like temperature and cpu load - not VPN tunnel details. Control of parameters is not possible.

Viprinet-Tunnel-SNMP monitors the status and performance of one VPN tunnel in a Viprinet Hub and not the Hub itself. There is one instance of this device to configured for each VPN tunnel to monitor. Control of parameters is not possible.

Viprinet-Rugged-Router-HTTPS monitors detailed status parameters of a Viprinet Router device, including the detailed state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface and also control of WiFi access point, Roaming and APN.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device

windows .

- --- The general status parameters of the device
- --- The status parameter of the interface slots / WAN interfaces of the device.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The Viprinet hub / router devices use 'public' as read community by default.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.mode	TEXT	R/O	StringRange R/O
info.modeMsg	TEXT	R/O	StringRange R/O
info.uptime	TEXT	R/O	StringRange R/O
info.firmwareStatus	TEXT	R/O	StringRange R/O
config.systemTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.cpuTempThreshold	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
averageCPULoad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
memoryUsage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
systemTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cpuTemperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
fan.1.enabled	TEXT	R/O	StringRange R/O
fan.2.enabled	TEXT	R/O	StringRange R/O
fan.1.status	TEXT	R/O	StringRange R/O
fan.2.status	TEXT	R/O	StringRange R/O
backplaneSlots	INTEGER	R/O	IntegerRange R/O (0 .. 0)
slot.1.name	TEXT	R/O	StringRange R/O DIS
slot.1.enabled	TEXT	R/O	StringRange R/O DIS
slot.1.status	TEXT	R/O	StringRange R/O DIS
slot.1.network	TEXT	R/O	StringRange R/O DIS
slot.1.serviceType	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
slot.1.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.1.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.1.roaming	TEXT	R/O	StringRange R/O DIS
slot.2.name	TEXT	R/O	StringRange R/O DIS
slot.2.enabled	TEXT	R/O	StringRange R/O DIS
slot.2.status	TEXT	R/O	StringRange R/O DIS
slot.2.network	TEXT	R/O	StringRange R/O DIS
slot.2.serviceType	TEXT	R/O	StringRange R/O DIS
slot.2.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.2.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.2.roaming	TEXT	R/O	StringRange R/O DIS
slot.3.name	TEXT	R/O	StringRange R/O DIS
slot.3.enabled	TEXT	R/O	StringRange R/O DIS
slot.3.status	TEXT	R/O	StringRange R/O DIS
slot.3.network	TEXT	R/O	StringRange R/O DIS
slot.3.serviceType	TEXT	R/O	StringRange R/O DIS
slot.3.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.3.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.3.roaming	TEXT	R/O	StringRange R/O DIS
slot.4.name	TEXT	R/O	StringRange R/O DIS
slot.4.enabled	TEXT	R/O	StringRange R/O DIS
slot.4.status	TEXT	R/O	StringRange R/O DIS
slot.4.network	TEXT	R/O	StringRange R/O DIS
slot.4.serviceType	TEXT	R/O	StringRange R/O DIS
slot.4.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.4.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.4.roaming	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
slot.5.name	TEXT	R/O	StringRange R/O DIS
slot.5.enabled	TEXT	R/O	StringRange R/O DIS
slot.5.status	TEXT	R/O	StringRange R/O DIS
slot.5.network	TEXT	R/O	StringRange R/O DIS
slot.5.serviceType	TEXT	R/O	StringRange R/O DIS
slot.5.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.5.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.5.roaming	TEXT	R/O	StringRange R/O DIS
slot.6.name	TEXT	R/O	StringRange R/O DIS
slot.6.enabled	TEXT	R/O	StringRange R/O DIS
slot.6.status	TEXT	R/O	StringRange R/O DIS
slot.6.network	TEXT	R/O	StringRange R/O DIS
slot.6.serviceType	TEXT	R/O	StringRange R/O DIS
slot.6.signal	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
slot.6.bandInfo	TEXT	R/O	StringRange R/O DIS
slot.6.roaming	TEXT	R/O	StringRange R/O DIS
faults.01	ALARM	R/O	AlarmFlagRange R/O (Power supply)
faults.02	ALARM	R/O	AlarmFlagRange R/O (System temp)
faults.03	ALARM	R/O	AlarmFlagRange R/O (CPU temp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Fan 1)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Fan 2)

9.732 Viprinet-Tunnel-SNMP

This page describes the Viprinet-Tunnel-SNMP device driver and the device window for this

driver.

The Viprinet-Tunnel-SNMP driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB-SNMP monitors some general status parameters of a Viprinet Hub device like temperature and cpu load - not VPN tunnel details. Control of parameters is not possible.

Viprinet-Rugged-Router-SNMP monitors some general status parameters of a Viprinet Router device, including the general state of the router's WAN interfaces. Control of parameters is not possible.

Viprinet-Rugged-Router-HTTPS monitors detailed status parameters of a Viprinet Router device, including the detailed state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface and also control of WiFi access point, Roaming and APN.

The Viprinet Hub, Tunnel and Router SNMP device drivers are monitoring only. A Viprinet Router is controlled over the HTTPS device driver and the Viprinet-HTTPS protocol.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The status parameters of the VPN tunnel
- --- The status parameters of up to six channels associated with this tunnel.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted,

the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

The Viprinet hub / router devices use 'public' as read community by default.

- tunnelIndex --- The (zero based) index of the tunnel to monitor.
- channelIndex --- The (zero based) index of the single channel to monitor.

Please note that multiple device driver instances which are talking to the same device must be configured with the same interface. Also the Viprinet-HUB-SNMP device must be configured within the same interface.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
config.tunnelIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.channelIndex.1	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.channelIndex.2	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.channelIndex.3	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.channelIndex.4	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.channelIndex.5	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.channelIndex.6	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
internal.ls	INTEGER		IntegerRange DIS (0 .. 0)
name	TEXT	R/O	StringRange R/O
enabled	TEXT	R/O	StringRange R/O
status	TEXT	R/O	StringRange R/O
maxBwToWan	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
maxBwFromWan	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
currentBwToWan	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
currentBwFromWan	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
remoteSerial	TEXT	R/O	StringRange R/O
connectedChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.1.name	TEXT	R/O	StringRange R/O DIS
channel.1.enabled	TEXT	R/O	StringRange R/O DIS
channel.1.status	TEXT	R/O	StringRange R/O DIS
channel.1.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.1.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.1.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.2.name	TEXT	R/O	StringRange R/O DIS
channel.2.enabled	TEXT	R/O	StringRange R/O DIS
channel.2.status	TEXT	R/O	StringRange R/O DIS
channel.2.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.2.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.2.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.3.name	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
channel.3.enabled	TEXT	R/O	StringRange R/O DIS
channel.3.status	TEXT	R/O	StringRange R/O DIS
channel.3.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.3.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.3.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.4.name	TEXT	R/O	StringRange R/O DIS
channel.4.enabled	TEXT	R/O	StringRange R/O DIS
channel.4.status	TEXT	R/O	StringRange R/O DIS
channel.4.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.4.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.4.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.5.name	TEXT	R/O	StringRange R/O DIS
channel.5.enabled	TEXT	R/O	StringRange R/O DIS
channel.5.status	TEXT	R/O	StringRange R/O DIS
channel.5.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.5.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.5.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.6.name	TEXT	R/O	StringRange R/O DIS
channel.6.enabled	TEXT	R/O	StringRange R/O DIS
channel.6.status	TEXT	R/O	StringRange R/O DIS
channel.6.up	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
channel.6.packetLoss	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
channel.6.linkStability	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tunnel disabled)

9.733 Viprinet-Tunnel

This page describes the Viprinet-Tunnel device driver and the device window for this driver.

The Viprinet-Tunnel driver is part of a group of device drivers used to monitor the Viprinet Hub / Router device family and thereby taking into account the modular concept of these devices:

Viprinet-HUB monitors some general status parameters of a Viprinet Hub device.

Viprinet-Router monitors some general status parameters of a Viprinet Router device, including the state of the router's WAN interfaces. It permits to reconnect / reset each individual WAN interface.

Viprinet-Tunnel monitors the status and performance of one VPN tunnel in a Viprinet Hub. There is one instance of this device to configured for each VPN tunnel to monitor.

All three device drivers use the ssh command line interface of the Viprinet device for monitoring and control.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The status parameters of the VPN tunnel
- --- The status parameters of up to six channels associated with this tunnel.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The ssh address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp username password*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts ssh connections. The port number may be omitted, the driver uses the standard port 22 in this case.
  - *username* --- The user name to log in at the device. The user name is mandatory.
  - *password* --- The password to login at the device. This is mandatory as well.

The Viprinet hub / router devices use 'root' as username and 'viprinet' as password by default.

- tunnelIndex --- The (zero based) index of the tunnel to monitor.

Please note that multiple device driver instances which are talking to the same device must be

configured with the same interface.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.routerName	TEXT	R/O	StringRange R/O
config.tunnelIndex	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
remoteSerial	TEXT	R/O	StringRange R/O
connectionStatus	TEXT	R/O	StringRange R/O
tunnelName	TEXT	R/O	StringRange R/O
connectedChannels	INTEGER	R/O	IntegerRange R/O (0 .. 0)
totReliableBwToWAN	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
totReliableBwFromWAN	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tunnelEnabled	CHOICE		EnumRange (YES NO)
tunnelReconnect	TEXT	NOPRESET	StringRange
channel.1.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.2.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.3.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.4.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.5.configured	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
channel.6.configured	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.1.name	TEXT		StringRange
channel.1.connectionStatus	TEXT	R/O	StringRange R/O
channel.1.enabled	TEXT	R/O	StringRange R/O
channel.1.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.1.lastError	TEXT	R/O	StringRange R/O
channel.1.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.1.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.1.reconnect	TEXT	NOPRESET	StringRange
channel.2.name	TEXT		StringRange
channel.2.connectionStatus	TEXT	R/O	StringRange R/O
channel.2.enabled	TEXT	R/O	StringRange R/O
channel.2.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.2.lastError	TEXT	R/O	StringRange R/O
channel.2.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.2.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.2.reconnect	TEXT	NOPRESET	StringRange
channel.3.name	TEXT		StringRange
channel.3.connectionStatus	TEXT	R/O	StringRange R/O
channel.3.enabled	TEXT	R/O	StringRange R/O
channel.3.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.3.lastError	TEXT	R/O	StringRange R/O
channel.3.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.3.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.3.reconnect	TEXT	NOPRESET	StringRange
channel.4.name	TEXT		StringRange

name	*type*	*flags*	*range*
channel.4.connectionStatus	TEXT	R/O	StringRange R/O
channel.4.enabled	TEXT	R/O	StringRange R/O
channel.4.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.4.lastError	TEXT	R/O	StringRange R/O
channel.4.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.4.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.4.reconnect	TEXT	NOPRESET	StringRange
channel.5.name	TEXT		StringRange
channel.5.connectionStatus	TEXT	R/O	StringRange R/O
channel.5.enabled	TEXT	R/O	StringRange R/O
channel.5.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.5.lastError	TEXT	R/O	StringRange R/O
channel.5.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.5.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.5.reconnect	TEXT	NOPRESET	StringRange
channel.6.name	TEXT		StringRange
channel.6.connectionStatus	TEXT	R/O	StringRange R/O
channel.6.enabled	TEXT	R/O	StringRange R/O
channel.6.status	BOOLEAN	R/O	BooleanRange R/O (true false)
channel.6.lastError	TEXT	R/O	StringRange R/O
channel.6.packetLoss	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.6.linkStability	INTEGER	R/O	IntegerRange R/O (0 .. 0)
channel.6.reconnect	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Tunnel index)

9.734 Virtual-Matrix-128x128

The *Virtual-Matrix-128x128* logical device extends the capabilities of a matrix with up to 128 output to switch ports by user defined signal names rather than by port numbers and permits to define sub-matrices.

Like with its smaller sibling, the Virtual-Matrix-128x32 , there is support for cascading matrices through an additional input for each matrix row. This way the *Virtual-Matrix-128x128* device may manage up to four cascaded SatService L-band switch matrices as one virtual matrix.

As the *Virtual-Matrix-128x128* supports up to 128 output ports, it uses 3-digit indexes for all 'output.???' and 'byname.???' variables.

Naming matrix inputs and outputs

With the web-based user interface of the SatService switch matrix devices you may define your own names for each particular input or output. The *Virtual-Matrix-128x128* device reads these names and uses them in the following way:

- The names of the outputs of the primary switch matrix are read and resebled as read-only variables 'output.name.???' in the same way the physical matrix does.
- The names of all physical inputs are read and used to build choice parameters called 'byname.???' for each output. You may use these choice parameters in user defined screens to let the operator select the signal by name rather than by input number.

The latter feature requires, that all matrix inputs are named uniquely. The *Virtual-Matrix-128x128* device detects ambiguous input names and rises the 'fault.05' (Duplicate input name). For use with the drop-down boxes of the 'byname.???' parameters the device renames ambiguous input names.

Defining sub-matrices

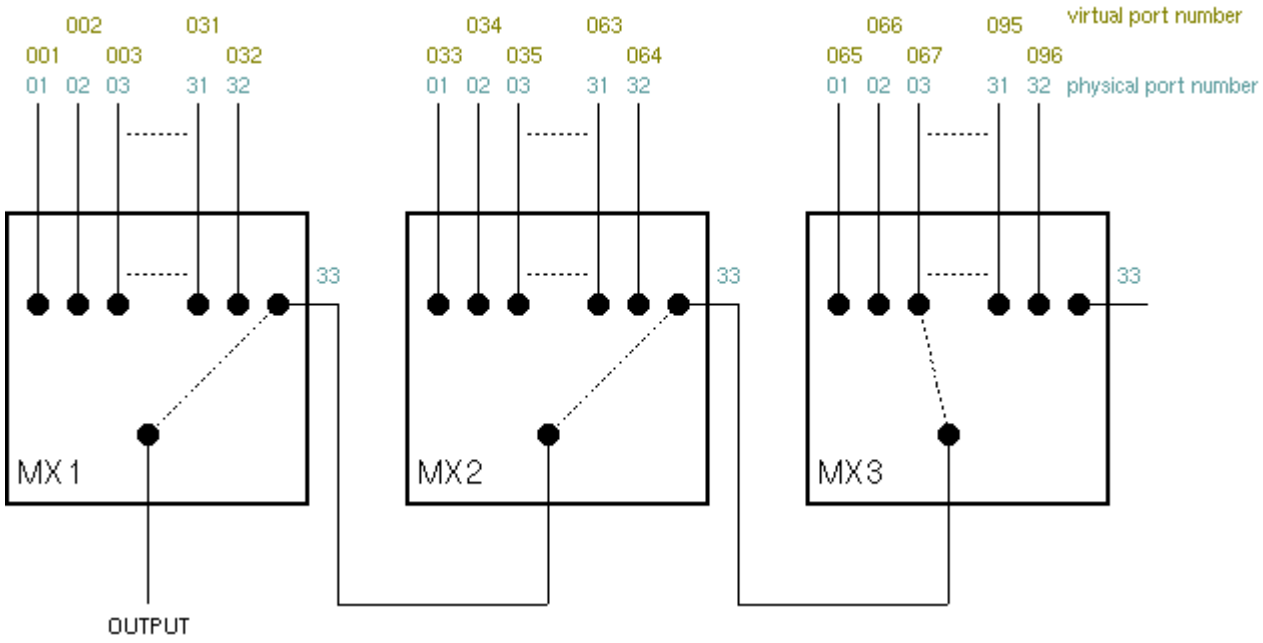
The *Virtual-Matrix-128x128* device lets you restrain the input/output ports of the matrix to a certain set of ports. This feature makes the virtual matrix device valuable even for a single physical switch matrix, without using the cascading capabilities. Several virtual matrix devices might be defined, each controlling another subset of the same physical switch matrix.

The device setup page lets you define a couple of parameters which each hold a list of permitted port numbers. If these lists are not empty, the virtual matrix device does not use all available port numbers but the set of ports defined in the particular list. The ports are re-numbered to a consecutive list of virtual port numbers.

Using this feature means, that by switching matrix ports on the physical layer, inputs may be accessed which are not present in the virtual matrix configuration. In this case, the virtual matrix shows this output to be connected to the matrix port '000' (named 'NO SIGNAL').

Cascading matrices

The SatService L-band switch matrices provide a cascading input port for each matrix output. This cascading input lets you extend the number of matrix inputs up to 128 (four matrices with 32 inputs each). Internally a matrix output may be switched to the cascading input using the input port number '33'.



The diagram above illustrates the signal routing and input number assignment for three cascaded 32-input matrices. The matrix MX1 - called the primary matrix - serves the (virtual) input ports 001..032, MX2 serves the ports 033 .. 064 and the last matrix MX3 serves the remaining input ports 065 .. 096.

This works with smaller matrices as well, even with matrices which provide a different number of inputs. The only requirement is, that all cascaded matrices are equipped with the same number of outputs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- abc --- Lets you select the input ports by name. The 128 possible output ports of the matrix are split over 4 pages, showing the selections for Ports 001-032, 033-064, 065-096 and 097-128.
- 1 2 3 --- Lets you select the input ports by port number. The 128 possible output ports of the matrix are split over two pages, showing the selections for Ports 001-064 and 065-128.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the*Virtual-Matrix-128x128* device to the devices it monitors / controls. Unused parameters should be left empty.

parameter	*description*
matrices	The list of physical matrices which shall be controlled by this device. This is a list of device names, separated by commas. The first device name identifies the primary matrix (that one providing the physical outputs), the other device names identify the subsequent matrices in the order of cascading. Example: **MX1,MX2,MX3**
cascadePort	This parmeter defines the input port to be used for cascading.
inputs.1	If this parameter is not empty, it defines the list of input ports which shall be accessible from the first matrix. Example: **{01,02,03,07,08}**
inputs.2	If this parameter is not empty, it defines the list of input ports which shall be accessible from the second matrix.
inputs.3	If this parameter is not empty, it defines the list of input ports which shall be accessible from the third matrix.
inputs.4	If this parameter is not empty, it defines the list of input ports which shall be accessible from the fourth matrix.
outputs	If this parameter is not empty, it defines the list of output ports which shall be accessible in the virtual matrix Example: **{001,002,003}**

Faults

The *Virtual-Matrix-128x32* device defines the following fault flags:

- *faults.01* --- Summary fault of the first (primary) physical matrix device.
- *faults.02* --- Summary fault of the second physical matrix device.
- *faults.03* --- Summary fault of the third physical matrix device.
- *faults.04* --- Summary fault of the fourth physical matrix device.
- *faults.05* --- Indicates that the physical matrices define duplicate input port names. The device *Virtual-Matrix-128x128* renamed the duplicate named ports for the usage in the 'byname.??' choice parameters. You may ignore this fault (i.e. set it's priority to OFF) if you do not intend to use the 'byname.??' port selection. For a meaningful usage of input names, you should rename the input ports at the physical matrix devices.

Remarks

Whenever you change the list of matrices or one of the lists matrix ports to be used in a sub-matrix in the device setup, you have to reset the *Virtual-Matrix-128x128* device by pressing the RESET button at maintenance page in order to make the changes effective. The same applies if you change input or output names at a physical matrix device.

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.matrices	TEXT	SAVE SETUP	StringRange
config.cascadePort	TEXT	SAVE SETUP	StringRange
config.inputs.1	TEXT	SAVE SETUP	StringRange
config.inputs.2	TEXT	SAVE SETUP	StringRange
config.inputs.3	TEXT	SAVE SETUP	StringRange
config.inputs.4	TEXT	SAVE SETUP	StringRange
config.outputs	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
output.001	CHOICE		EnumRange DIS (???)
output.002	CHOICE		EnumRange DIS (???)
output.003	CHOICE		EnumRange DIS (???)
output.004	CHOICE		EnumRange DIS (???)
output.005	CHOICE		EnumRange DIS (???)
output.006	CHOICE		EnumRange DIS (???)
output.007	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.008	CHOICE		EnumRange DIS (???)
output.009	CHOICE		EnumRange DIS (???)
output.010	CHOICE		EnumRange DIS (???)
output.011	CHOICE		EnumRange DIS (???)
output.012	CHOICE		EnumRange DIS (???)
output.013	CHOICE		EnumRange DIS (???)
output.014	CHOICE		EnumRange DIS (???)
output.015	CHOICE		EnumRange DIS (???)
output.016	CHOICE		EnumRange DIS (???)
output.017	CHOICE		EnumRange DIS (???)
output.018	CHOICE		EnumRange DIS (???)
output.019	CHOICE		EnumRange DIS (???)
output.020	CHOICE		EnumRange DIS (???)
output.021	CHOICE		EnumRange DIS (???)
output.022	CHOICE		EnumRange DIS (???)
output.023	CHOICE		EnumRange DIS (???)
output.024	CHOICE		EnumRange DIS (???)
output.025	CHOICE		EnumRange DIS (???)
output.026	CHOICE		EnumRange DIS (???)
output.027	CHOICE		EnumRange DIS (???)
output.028	CHOICE		EnumRange DIS (???)
output.029	CHOICE		EnumRange DIS (???)
output.030	CHOICE		EnumRange DIS (???)
output.031	CHOICE		EnumRange DIS (???)
output.032	CHOICE		EnumRange DIS (???)
output.033	CHOICE		EnumRange DIS (???)
output.034	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.035	CHOICE		EnumRange DIS (???)
output.036	CHOICE		EnumRange DIS (???)
output.037	CHOICE		EnumRange DIS (???)
output.038	CHOICE		EnumRange DIS (???)
output.039	CHOICE		EnumRange DIS (???)
output.040	CHOICE		EnumRange DIS (???)
output.041	CHOICE		EnumRange DIS (???)
output.042	CHOICE		EnumRange DIS (???)
output.043	CHOICE		EnumRange DIS (???)
output.044	CHOICE		EnumRange DIS (???)
output.045	CHOICE		EnumRange DIS (???)
output.046	CHOICE		EnumRange DIS (???)
output.047	CHOICE		EnumRange DIS (???)
output.048	CHOICE		EnumRange DIS (???)
output.049	CHOICE		EnumRange DIS (???)
output.050	CHOICE		EnumRange DIS (???)
output.051	CHOICE		EnumRange DIS (???)
output.052	CHOICE		EnumRange DIS (???)
output.053	CHOICE		EnumRange DIS (???)
output.054	CHOICE		EnumRange DIS (???)
output.055	CHOICE		EnumRange DIS (???)
output.056	CHOICE		EnumRange DIS (???)
output.057	CHOICE		EnumRange DIS (???)
output.058	CHOICE		EnumRange DIS (???)
output.059	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.060	CHOICE		EnumRange DIS (???)
output.061	CHOICE		EnumRange DIS (???)
output.062	CHOICE		EnumRange DIS (???)
output.063	CHOICE		EnumRange DIS (???)
output.064	CHOICE		EnumRange DIS (???)
output.065	CHOICE		EnumRange DIS (???)
output.066	CHOICE		EnumRange DIS (???)
output.067	CHOICE		EnumRange DIS (???)
output.068	CHOICE		EnumRange DIS (???)
output.069	CHOICE		EnumRange DIS (???)
output.070	CHOICE		EnumRange DIS (???)
output.071	CHOICE		EnumRange DIS (???)
output.072	CHOICE		EnumRange DIS (???)
output.073	CHOICE		EnumRange DIS (???)
output.074	CHOICE		EnumRange DIS (???)
output.075	CHOICE		EnumRange DIS (???)
output.076	CHOICE		EnumRange DIS (???)
output.077	CHOICE		EnumRange DIS (???)
output.078	CHOICE		EnumRange DIS (???)
output.079	CHOICE		EnumRange DIS (???)
output.080	CHOICE		EnumRange DIS (???)
output.081	CHOICE		EnumRange DIS (???)
output.082	CHOICE		EnumRange DIS (???)
output.083	CHOICE		EnumRange DIS (???)
output.084	CHOICE		EnumRange DIS (???)
output.085	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.086	CHOICE		EnumRange DIS (???)
output.087	CHOICE		EnumRange DIS (???)
output.088	CHOICE		EnumRange DIS (???)
output.089	CHOICE		EnumRange DIS (???)
output.090	CHOICE		EnumRange DIS (???)
output.091	CHOICE		EnumRange DIS (???)
output.092	CHOICE		EnumRange DIS (???)
output.093	CHOICE		EnumRange DIS (???)
output.094	CHOICE		EnumRange DIS (???)
output.095	CHOICE		EnumRange DIS (???)
output.096	CHOICE		EnumRange DIS (???)
output.097	CHOICE		EnumRange DIS (???)
output.098	CHOICE		EnumRange DIS (???)
output.099	CHOICE		EnumRange DIS (???)
output.100	CHOICE		EnumRange DIS (???)
output.101	CHOICE		EnumRange DIS (???)
output.102	CHOICE		EnumRange DIS (???)
output.103	CHOICE		EnumRange DIS (???)
output.104	CHOICE		EnumRange DIS (???)
output.105	CHOICE		EnumRange DIS (???)
output.106	CHOICE		EnumRange DIS (???)
output.107	CHOICE		EnumRange DIS (???)
output.108	CHOICE		EnumRange DIS (???)
output.109	CHOICE		EnumRange DIS (???)
output.110	CHOICE		EnumRange DIS (???)
output.111	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.112	CHOICE		EnumRange DIS (???)
output.113	CHOICE		EnumRange DIS (???)
output.114	CHOICE		EnumRange DIS (???)
output.115	CHOICE		EnumRange DIS (???)
output.116	CHOICE		EnumRange DIS (???)
output.117	CHOICE		EnumRange DIS (???)
output.118	CHOICE		EnumRange DIS (???)
output.119	CHOICE		EnumRange DIS (???)
output.120	CHOICE		EnumRange DIS (???)
output.121	CHOICE		EnumRange DIS (???)
output.122	CHOICE		EnumRange DIS (???)
output.123	CHOICE		EnumRange DIS (???)
output.124	CHOICE		EnumRange DIS (???)
output.125	CHOICE		EnumRange DIS (???)
output.126	CHOICE		EnumRange DIS (???)
output.127	CHOICE		EnumRange DIS (???)
output.128	CHOICE		EnumRange DIS (???)
byname.001	CHOICE		EnumRange DIS (?)
byname.002	CHOICE		EnumRange DIS (?)
byname.003	CHOICE		EnumRange DIS (?)
byname.004	CHOICE		EnumRange DIS (?)
byname.005	CHOICE		EnumRange DIS (?)
byname.006	CHOICE		EnumRange DIS (?)
byname.007	CHOICE		EnumRange DIS (?)
byname.008	CHOICE		EnumRange DIS (?)
byname.009	CHOICE		EnumRange DIS (?)
byname.010	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.011	CHOICE		EnumRange DIS (?)
byname.012	CHOICE		EnumRange DIS (?)
byname.013	CHOICE		EnumRange DIS (?)
byname.014	CHOICE		EnumRange DIS (?)
byname.015	CHOICE		EnumRange DIS (?)
byname.016	CHOICE		EnumRange DIS (?)
byname.017	CHOICE		EnumRange DIS (?)
byname.018	CHOICE		EnumRange DIS (?)
byname.019	CHOICE		EnumRange DIS (?)
byname.020	CHOICE		EnumRange DIS (?)
byname.021	CHOICE		EnumRange DIS (?)
byname.022	CHOICE		EnumRange DIS (?)
byname.023	CHOICE		EnumRange DIS (?)
byname.024	CHOICE		EnumRange DIS (?)
byname.025	CHOICE		EnumRange DIS (?)
byname.026	CHOICE		EnumRange DIS (?)
byname.027	CHOICE		EnumRange DIS (?)
byname.028	CHOICE		EnumRange DIS (?)
byname.029	CHOICE		EnumRange DIS (?)
byname.030	CHOICE		EnumRange DIS (?)
byname.031	CHOICE		EnumRange DIS (?)
byname.032	CHOICE		EnumRange DIS (?)
byname.033	CHOICE		EnumRange DIS (?)
byname.034	CHOICE		EnumRange DIS (?)
byname.035	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.036	CHOICE		EnumRange DIS (?)
byname.037	CHOICE		EnumRange DIS (?)
byname.038	CHOICE		EnumRange DIS (?)
byname.039	CHOICE		EnumRange DIS (?)
byname.040	CHOICE		EnumRange DIS (?)
byname.041	CHOICE		EnumRange DIS (?)
byname.042	CHOICE		EnumRange DIS (?)
byname.043	CHOICE		EnumRange DIS (?)
byname.044	CHOICE		EnumRange DIS (?)
byname.045	CHOICE		EnumRange DIS (?)
byname.046	CHOICE		EnumRange DIS (?)
byname.047	CHOICE		EnumRange DIS (?)
byname.048	CHOICE		EnumRange DIS (?)
byname.049	CHOICE		EnumRange DIS (?)
byname.050	CHOICE		EnumRange DIS (?)
byname.051	CHOICE		EnumRange DIS (?)
byname.052	CHOICE		EnumRange DIS (?)
byname.053	CHOICE		EnumRange DIS (?)
byname.054	CHOICE		EnumRange DIS (?)
byname.055	CHOICE		EnumRange DIS (?)
byname.056	CHOICE		EnumRange DIS (?)
byname.057	CHOICE		EnumRange DIS (?)
byname.058	CHOICE		EnumRange DIS (?)
byname.059	CHOICE		EnumRange DIS (?)
byname.060	CHOICE		EnumRange DIS (?)
byname.061	CHOICE		EnumRange DIS (?)
byname.062	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.063	CHOICE		EnumRange DIS (?)
byname.064	CHOICE		EnumRange DIS (?)
byname.065	CHOICE		EnumRange DIS (?)
byname.066	CHOICE		EnumRange DIS (?)
byname.067	CHOICE		EnumRange DIS (?)
byname.068	CHOICE		EnumRange DIS (?)
byname.069	CHOICE		EnumRange DIS (?)
byname.070	CHOICE		EnumRange DIS (?)
byname.071	CHOICE		EnumRange DIS (?)
byname.072	CHOICE		EnumRange DIS (?)
byname.073	CHOICE		EnumRange DIS (?)
byname.074	CHOICE		EnumRange DIS (?)
byname.075	CHOICE		EnumRange DIS (?)
byname.076	CHOICE		EnumRange DIS (?)
byname.077	CHOICE		EnumRange DIS (?)
byname.078	CHOICE		EnumRange DIS (?)
byname.079	CHOICE		EnumRange DIS (?)
byname.080	CHOICE		EnumRange DIS (?)
byname.081	CHOICE		EnumRange DIS (?)
byname.082	CHOICE		EnumRange DIS (?)
byname.083	CHOICE		EnumRange DIS (?)
byname.084	CHOICE		EnumRange DIS (?)
byname.085	CHOICE		EnumRange DIS (?)
byname.086	CHOICE		EnumRange DIS (?)
byname.087	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.088	CHOICE		EnumRange DIS (?)
byname.089	CHOICE		EnumRange DIS (?)
byname.090	CHOICE		EnumRange DIS (?)
byname.091	CHOICE		EnumRange DIS (?)
byname.092	CHOICE		EnumRange DIS (?)
byname.093	CHOICE		EnumRange DIS (?)
byname.094	CHOICE		EnumRange DIS (?)
byname.095	CHOICE		EnumRange DIS (?)
byname.096	CHOICE		EnumRange DIS (?)
byname.097	CHOICE		EnumRange DIS (?)
byname.098	CHOICE		EnumRange DIS (?)
byname.099	CHOICE		EnumRange DIS (?)
byname.100	CHOICE		EnumRange DIS (?)
byname.101	CHOICE		EnumRange DIS (?)
byname.102	CHOICE		EnumRange DIS (?)
byname.103	CHOICE		EnumRange DIS (?)
byname.104	CHOICE		EnumRange DIS (?)
byname.105	CHOICE		EnumRange DIS (?)
byname.106	CHOICE		EnumRange DIS (?)
byname.107	CHOICE		EnumRange DIS (?)
byname.108	CHOICE		EnumRange DIS (?)
byname.109	CHOICE		EnumRange DIS (?)
byname.110	CHOICE		EnumRange DIS (?)
byname.111	CHOICE		EnumRange DIS (?)
byname.112	CHOICE		EnumRange DIS (?)
byname.113	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.114	CHOICE		EnumRange DIS (?)
byname.115	CHOICE		EnumRange DIS (?)
byname.116	CHOICE		EnumRange DIS (?)
byname.117	CHOICE		EnumRange DIS (?)
byname.118	CHOICE		EnumRange DIS (?)
byname.119	CHOICE		EnumRange DIS (?)
byname.120	CHOICE		EnumRange DIS (?)
byname.121	CHOICE		EnumRange DIS (?)
byname.122	CHOICE		EnumRange DIS (?)
byname.123	CHOICE		EnumRange DIS (?)
byname.124	CHOICE		EnumRange DIS (?)
byname.125	CHOICE		EnumRange DIS (?)
byname.126	CHOICE		EnumRange DIS (?)
byname.127	CHOICE		EnumRange DIS (?)
byname.128	CHOICE		EnumRange DIS (?)
output.name.001	TEXT	R/O	StringRange R/O
output.name.002	TEXT	R/O	StringRange R/O
output.name.003	TEXT	R/O	StringRange R/O
output.name.004	TEXT	R/O	StringRange R/O
output.name.005	TEXT	R/O	StringRange R/O
output.name.006	TEXT	R/O	StringRange R/O
output.name.007	TEXT	R/O	StringRange R/O
output.name.008	TEXT	R/O	StringRange R/O
output.name.009	TEXT	R/O	StringRange R/O
output.name.010	TEXT	R/O	StringRange R/O
output.name.011	TEXT	R/O	StringRange R/O
output.name.012	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.013	TEXT	R/O	StringRange R/O
output.name.014	TEXT	R/O	StringRange R/O
output.name.015	TEXT	R/O	StringRange R/O
output.name.016	TEXT	R/O	StringRange R/O
output.name.017	TEXT	R/O	StringRange R/O
output.name.018	TEXT	R/O	StringRange R/O
output.name.019	TEXT	R/O	StringRange R/O
output.name.020	TEXT	R/O	StringRange R/O
output.name.021	TEXT	R/O	StringRange R/O
output.name.022	TEXT	R/O	StringRange R/O
output.name.023	TEXT	R/O	StringRange R/O
output.name.024	TEXT	R/O	StringRange R/O
output.name.025	TEXT	R/O	StringRange R/O
output.name.026	TEXT	R/O	StringRange R/O
output.name.027	TEXT	R/O	StringRange R/O
output.name.028	TEXT	R/O	StringRange R/O
output.name.029	TEXT	R/O	StringRange R/O
output.name.030	TEXT	R/O	StringRange R/O
output.name.031	TEXT	R/O	StringRange R/O
output.name.032	TEXT	R/O	StringRange R/O
output.name.033	TEXT	R/O	StringRange R/O
output.name.034	TEXT	R/O	StringRange R/O
output.name.035	TEXT	R/O	StringRange R/O
output.name.036	TEXT	R/O	StringRange R/O
output.name.037	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.038	TEXT	R/O	StringRange R/O
output.name.039	TEXT	R/O	StringRange R/O
output.name.040	TEXT	R/O	StringRange R/O
output.name.041	TEXT	R/O	StringRange R/O
output.name.042	TEXT	R/O	StringRange R/O
output.name.043	TEXT	R/O	StringRange R/O
output.name.044	TEXT	R/O	StringRange R/O
output.name.045	TEXT	R/O	StringRange R/O
output.name.046	TEXT	R/O	StringRange R/O
output.name.047	TEXT	R/O	StringRange R/O
output.name.048	TEXT	R/O	StringRange R/O
output.name.049	TEXT	R/O	StringRange R/O
output.name.050	TEXT	R/O	StringRange R/O
output.name.051	TEXT	R/O	StringRange R/O
output.name.052	TEXT	R/O	StringRange R/O
output.name.053	TEXT	R/O	StringRange R/O
output.name.054	TEXT	R/O	StringRange R/O
output.name.055	TEXT	R/O	StringRange R/O
output.name.056	TEXT	R/O	StringRange R/O
output.name.057	TEXT	R/O	StringRange R/O
output.name.058	TEXT	R/O	StringRange R/O
output.name.059	TEXT	R/O	StringRange R/O
output.name.060	TEXT	R/O	StringRange R/O
output.name.061	TEXT	R/O	StringRange R/O
output.name.062	TEXT	R/O	StringRange R/O
output.name.063	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.064	TEXT	R/O	StringRange R/O
output.name.065	TEXT	R/O	StringRange R/O
output.name.066	TEXT	R/O	StringRange R/O
output.name.067	TEXT	R/O	StringRange R/O
output.name.068	TEXT	R/O	StringRange R/O
output.name.069	TEXT	R/O	StringRange R/O
output.name.070	TEXT	R/O	StringRange R/O
output.name.071	TEXT	R/O	StringRange R/O
output.name.072	TEXT	R/O	StringRange R/O
output.name.073	TEXT	R/O	StringRange R/O
output.name.074	TEXT	R/O	StringRange R/O
output.name.075	TEXT	R/O	StringRange R/O
output.name.076	TEXT	R/O	StringRange R/O
output.name.077	TEXT	R/O	StringRange R/O
output.name.078	TEXT	R/O	StringRange R/O
output.name.079	TEXT	R/O	StringRange R/O
output.name.080	TEXT	R/O	StringRange R/O
output.name.081	TEXT	R/O	StringRange R/O
output.name.082	TEXT	R/O	StringRange R/O
output.name.083	TEXT	R/O	StringRange R/O
output.name.084	TEXT	R/O	StringRange R/O
output.name.085	TEXT	R/O	StringRange R/O
output.name.086	TEXT	R/O	StringRange R/O
output.name.087	TEXT	R/O	StringRange R/O
output.name.088	TEXT	R/O	StringRange R/O
output.name.089	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.090	TEXT	R/O	StringRange R/O
output.name.091	TEXT	R/O	StringRange R/O
output.name.092	TEXT	R/O	StringRange R/O
output.name.093	TEXT	R/O	StringRange R/O
output.name.094	TEXT	R/O	StringRange R/O
output.name.095	TEXT	R/O	StringRange R/O
output.name.096	TEXT	R/O	StringRange R/O
output.name.097	TEXT	R/O	StringRange R/O
output.name.098	TEXT	R/O	StringRange R/O
output.name.099	TEXT	R/O	StringRange R/O
output.name.100	TEXT	R/O	StringRange R/O
output.name.101	TEXT	R/O	StringRange R/O
output.name.102	TEXT	R/O	StringRange R/O
output.name.103	TEXT	R/O	StringRange R/O
output.name.104	TEXT	R/O	StringRange R/O
output.name.105	TEXT	R/O	StringRange R/O
output.name.106	TEXT	R/O	StringRange R/O
output.name.107	TEXT	R/O	StringRange R/O
output.name.108	TEXT	R/O	StringRange R/O
output.name.109	TEXT	R/O	StringRange R/O
output.name.110	TEXT	R/O	StringRange R/O
output.name.111	TEXT	R/O	StringRange R/O
output.name.112	TEXT	R/O	StringRange R/O
output.name.113	TEXT	R/O	StringRange R/O
output.name.114	TEXT	R/O	StringRange R/O
output.name.115	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.116	TEXT	R/O	StringRange R/O
output.name.117	TEXT	R/O	StringRange R/O
output.name.118	TEXT	R/O	StringRange R/O
output.name.119	TEXT	R/O	StringRange R/O
output.name.120	TEXT	R/O	StringRange R/O
output.name.121	TEXT	R/O	StringRange R/O
output.name.122	TEXT	R/O	StringRange R/O
output.name.123	TEXT	R/O	StringRange R/O
output.name.124	TEXT	R/O	StringRange R/O
output.name.125	TEXT	R/O	StringRange R/O
output.name.126	TEXT	R/O	StringRange R/O
output.name.127	TEXT	R/O	StringRange R/O
output.name.128	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix 1 Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Matrix 2 Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Matrix 3 Summary)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Matrix 4 Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Duplicate input name)

9.735 Virtual-Matrix-128x32

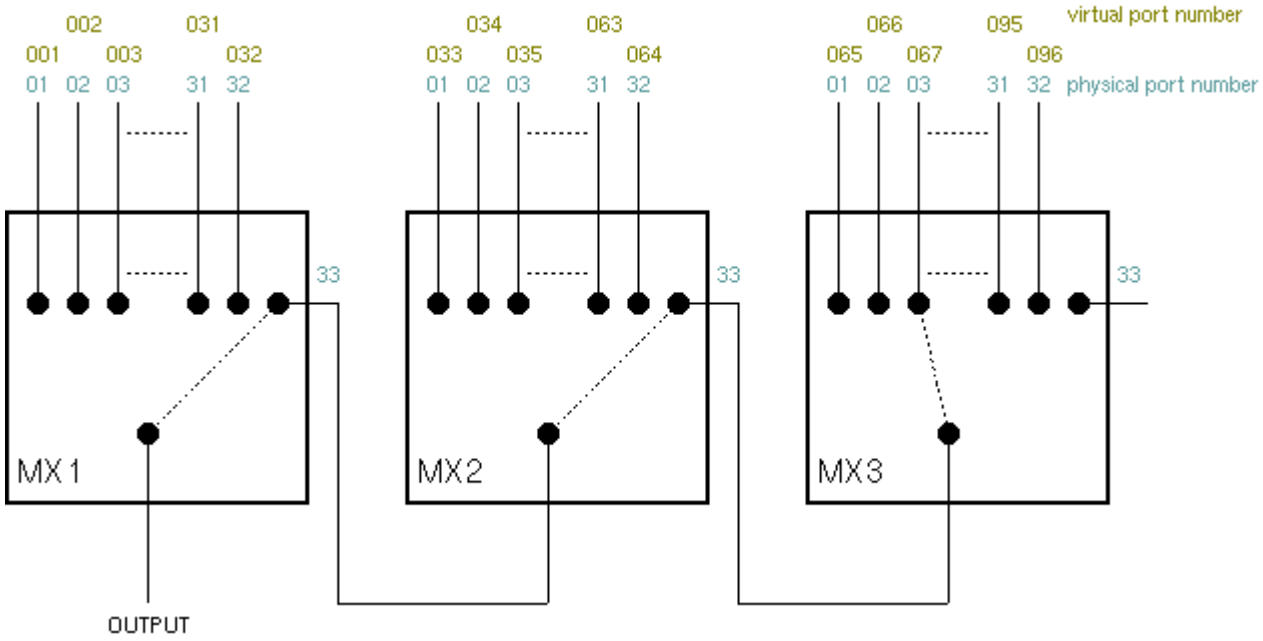
The *Virtual-Matrix-128x32* logical device lets you control up to four cascaded SatService L-band switch matrices as one virtual matrix. The *Virtual-Matrix-128x32* device is capable to handle partial equipped matrix devices down to 8x8 port matrices.

Beside the handling of cascaded matrices, the *Virtual-Matrix-128x32* device extends the SatService matrix driver by capabilities to define sub-matrices and to select input ports by their name rather than their number.

Cascading matrices

The SatService L-band switch matrices provide a cascading input port for each matrix output.

This cascading input lets you extend the number of matrix inputs up to 128 (four matrices with 32 inputs each). Internally a matrix output may be switched to the cascading input using the input port number '33'.



The diagram above illustrates the signal routing and input number assignment for three cascaded 32-input matrices. The matrix MX1 - called the primary matrix - serves the (virtual) input ports 001..032, MX2 serves the ports 033 .. 064 and the last matrix MX3 serves the remaining input ports 065 .. 096.

This works with smaller matrices as well, even with matrices which provide a different number of inputs. The only requirement is, that all cascaded matrices are equipped with the same number of outputs.

Naming matrix inputs and outputs

With the web-based user interface of the SatService switch matrix devices you may define your own names for each particular input or output. The *Virtual-Matrix-128x32* device reads these names and uses them in the following way:

- The names of the outputs of the primary switch matrix are read and resebled as read-only variables 'output.name.??' in the same way the physical matrix does.
- The names of all physical inputs are read and used to build choice parameters called 'byname.??' for each output. You may use these choice parameters in user defined screens to let the operator select the signal by name rather than by input number.

The latter feature requires, that all matrix inputs are named uniquely. The *Virtual-Matrix-128x32* device detects ambiguous input names and rises the 'fault.05' (Duplicate input name). For use with the drop-down boxes of the 'byname.??' parameters the device renames ambiguous input names.

Defining sub-matrices

The *Virtual-Matrix-128x32* device lets you restrain the input/output ports of the matrix to a certain set of ports. This feature makes the virtual matrix device valuable even for a single physical switch matrix, without using the cascading capabilities. Several virtual matrix devices might be defined, each controlling another subset of the same physical switch matrix.

The device setup page lets you define a couple of parameters which each hold a list of permitted port numbers. If these lists are not empty, the virtual matrix device does not use all available port numbers but the set of ports defined in the particular list. The ports are re-numbered to a consecutive list of virtual port numbers.

Using this feature means, that by switching matrix ports on the physical layer, inputs may be accessed which are not present in the virtual matrix configuration. In this case, the virtual matrix shows this output to be connected to the matrix port '000' (named 'NO SIGNAL').

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Matrix port select
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

The table below explains the configuration parameters which interconnect the*Virtual-Matrix-128x32* device to the devices it monitors / controls. Unused parameters should be left empty.

parameter	***description***
matrices	The list of physical matrices which shall be controlled by this device. This is a list of device names, separated by commas. The first device name identifies the primary matrix (that one providing the physical outputs), the other device names identify the subsequent matrices in the order of cascading. Example: **MX1,MX2,MX3**
inputs.1	If this parameter is not empty, it defines the list of input ports which shall be accessible from the first matrix. Example: **{01,02,03,07,08}**
inputs.2	If this parameter is not empty, it defines the list of input ports which shall be accessible from the second matrix.
inputs.3	If this parameter is not empty, it defines the list of input ports which shall be accessible from the third matrix.

parameter	*description*
inputs.4	If this parameter is not empty, it defines the list of input ports which shall be accessible from the fourth matrix.
outputs	If this parameter is not empty, it defines the list of output ports which shall be accessible in the virtual matrix Example: **{01,02,03}**

Faults

The *Virtual-Matrix-128x32* device defines the following fault flags:

- *faults.01* --- Summary fault of the first (primary) physical matrix device.
- *faults.02* --- Summary fault of the second physical matrix device.
- *faults.03* --- Summary fault of the third physical matrix device.
- *faults.04* --- Summary fault of the fourth physical matrix device.
- *faults.05* --- Indicates that the physical matrices define duplicate input port names. The device *Virtual-Matrix-128x32* renamed the duplicate named ports for the usage in the 'byname.??' choice parameters. You may ignore this fault (i.e. set it's priority to OFF) if you do not intend to use the 'byname.??' port selection. For a meaningful usage of input names, you should rename the input ports at the physical matrix devices.

Remarks

Whenever you change the list of matrices or one of the lists matrix ports to be used in a sub-matrix in the device setup, you have to reset the *Virtual-Matrix-128x32* device by pressing the RESET button at maintenance page in order to make the changes effective. The same applies if you change input or output names at a physical matrix device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.inputs	TEXT	R/O	StringRange R/O
info.outputs	TEXT	R/O	StringRange R/O
info.state	TEXT	R/O	StringRange R/O
config.matrices	TEXT	SAVE SETUP	StringRange
config.inputs.1	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.inputs.2	TEXT	SAVE SETUP	StringRange
config.inputs.3	TEXT	SAVE SETUP	StringRange
config.inputs.4	TEXT	SAVE SETUP	StringRange
config.outputs	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
output.01	CHOICE		EnumRange DIS (???)
output.02	CHOICE		EnumRange DIS (???)
output.03	CHOICE		EnumRange DIS (???)
output.04	CHOICE		EnumRange DIS (???)
output.05	CHOICE		EnumRange DIS (???)
output.06	CHOICE		EnumRange DIS (???)
output.07	CHOICE		EnumRange DIS (???)
output.08	CHOICE		EnumRange DIS (???)
output.09	CHOICE		EnumRange DIS (???)
output.10	CHOICE		EnumRange DIS (???)
output.11	CHOICE		EnumRange DIS (???)
output.12	CHOICE		EnumRange DIS (???)
output.13	CHOICE		EnumRange DIS (???)
output.14	CHOICE		EnumRange DIS (???)
output.15	CHOICE		EnumRange DIS (???)
output.16	CHOICE		EnumRange DIS (???)
output.17	CHOICE		EnumRange DIS (???)
output.18	CHOICE		EnumRange DIS (???)
output.19	CHOICE		EnumRange DIS (???)
output.20	CHOICE		EnumRange DIS (???)
output.21	CHOICE		EnumRange DIS (???)
output.22	CHOICE		EnumRange DIS (???)

name	*type*	*flags*	*range*
output.23	CHOICE		EnumRange DIS (???)
output.24	CHOICE		EnumRange DIS (???)
output.25	CHOICE		EnumRange DIS (???)
output.26	CHOICE		EnumRange DIS (???)
output.27	CHOICE		EnumRange DIS (???)
output.28	CHOICE		EnumRange DIS (???)
output.29	CHOICE		EnumRange DIS (???)
output.30	CHOICE		EnumRange DIS (???)
output.31	CHOICE		EnumRange DIS (???)
output.32	CHOICE		EnumRange DIS (???)
byname.01	CHOICE		EnumRange DIS (?)
byname.02	CHOICE		EnumRange DIS (?)
byname.03	CHOICE		EnumRange DIS (?)
byname.04	CHOICE		EnumRange DIS (?)
byname.05	CHOICE		EnumRange DIS (?)
byname.06	CHOICE		EnumRange DIS (?)
byname.07	CHOICE		EnumRange DIS (?)
byname.08	CHOICE		EnumRange DIS (?)
byname.09	CHOICE		EnumRange DIS (?)
byname.10	CHOICE		EnumRange DIS (?)
byname.11	CHOICE		EnumRange DIS (?)
byname.12	CHOICE		EnumRange DIS (?)
byname.13	CHOICE		EnumRange DIS (?)
byname.14	CHOICE		EnumRange DIS (?)
byname.15	CHOICE		EnumRange DIS (?)

name	*type*	*flags*	*range*
byname.16	CHOICE		EnumRange DIS (?)
byname.17	CHOICE		EnumRange DIS (?)
byname.18	CHOICE		EnumRange DIS (?)
byname.19	CHOICE		EnumRange DIS (?)
byname.20	CHOICE		EnumRange DIS (?)
byname.21	CHOICE		EnumRange DIS (?)
byname.22	CHOICE		EnumRange DIS (?)
byname.23	CHOICE		EnumRange DIS (?)
byname.24	CHOICE		EnumRange DIS (?)
byname.25	CHOICE		EnumRange DIS (?)
byname.26	CHOICE		EnumRange DIS (?)
byname.27	CHOICE		EnumRange DIS (?)
byname.28	CHOICE		EnumRange DIS (?)
byname.29	CHOICE		EnumRange DIS (?)
byname.30	CHOICE		EnumRange DIS (?)
byname.31	CHOICE		EnumRange DIS (?)
byname.32	CHOICE		EnumRange DIS (?)
output.name.01	TEXT	R/O	StringRange R/O
output.name.02	TEXT	R/O	StringRange R/O
output.name.03	TEXT	R/O	StringRange R/O
output.name.04	TEXT	R/O	StringRange R/O
output.name.05	TEXT	R/O	StringRange R/O
output.name.06	TEXT	R/O	StringRange R/O
output.name.07	TEXT	R/O	StringRange R/O
output.name.08	TEXT	R/O	StringRange R/O
output.name.09	TEXT	R/O	StringRange R/O
output.name.10	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
output.name.11	TEXT	R/O	StringRange R/O
output.name.12	TEXT	R/O	StringRange R/O
output.name.13	TEXT	R/O	StringRange R/O
output.name.14	TEXT	R/O	StringRange R/O
output.name.15	TEXT	R/O	StringRange R/O
output.name.16	TEXT	R/O	StringRange R/O
output.name.17	TEXT	R/O	StringRange R/O
output.name.18	TEXT	R/O	StringRange R/O
output.name.19	TEXT	R/O	StringRange R/O
output.name.20	TEXT	R/O	StringRange R/O
output.name.21	TEXT	R/O	StringRange R/O
output.name.22	TEXT	R/O	StringRange R/O
output.name.23	TEXT	R/O	StringRange R/O
output.name.24	TEXT	R/O	StringRange R/O
output.name.25	TEXT	R/O	StringRange R/O
output.name.26	TEXT	R/O	StringRange R/O
output.name.27	TEXT	R/O	StringRange R/O
output.name.28	TEXT	R/O	StringRange R/O
output.name.29	TEXT	R/O	StringRange R/O
output.name.30	TEXT	R/O	StringRange R/O
output.name.31	TEXT	R/O	StringRange R/O
output.name.32	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Matrix 1 Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Matrix 2 Summary)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Matrix 3 Summary)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Matrix 4 Summary)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Duplicate input name)

9.736 Vitec-MGW-Encoder

This page describes the device driver and the device window for the Vitec MGW Encoder using REST HTTPs.

It was developed and tested with the following SW releases:

- **1.4.0 rc1**
- **1.5.0**

It supports the primary channel configuration of one or two IP outputs which can be using SRT, UDP, RTP, RTMP or RTSP protocol.

The driver can only handle one encoder / channel instance. As The Vitec can handle up to 4 primary channels. Each encoder/channel has to be setup in a single device. The respective instance has to be configured at first setup via config.encoder.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains all MPEG output parameters.
- --- Contains IP output settings and statistics.
- --- Contains the video encoder settings.
- --- Contains Audio Parameters for up to 8 stereo channels.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Vitec MGW Encoder.

parameter	*description*

parameter	*description*
address	The Management IP address specification for the device. If authentification is enabled username and password have to be added. When using more than one device/route, the address can refer to existing device via @DeviceName
encoder	Defines the desired encoder/channel instance.
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.productName	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.boardSerial	TEXT	R/O	StringRange R/O
info.hwFpga	TEXT	R/O	StringRange R/O
info.hwVideo	TEXT	R/O	StringRange R/O
info.hwSystem	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.chipTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.encoder	CHOICE	SAVE SETUP	EnumRange (1 2 3 4)
config.readBackDelay	INTEGER	SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
opState	CHOICE		EnumRange (START STOP)
video.enable	CHOICE		EnumRange (ON OFF)
video.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 100.000)
video.maxrate	FLOAT	R/O	DoubleRange R/O (0.000 .. 100.000)
video.codec	CHOICE	R/O	EnumRange R/O (H264 HEVC MPEG2)
video.profile	CHOICE		EnumRange (H264-BASELINE H264-MAIN H264-HIGH H264-HIGH-10 H264-HIGH-4:2:2 HEVC-8-Bit-4:2:0 HEVC-8-Bit-4:2:2 HEVC-10-Bit-4:2:0 HEVC-10-Bit-4:2:2 MPEG2-8bit-4:2:0)
video.latency	CHOICE		EnumRange (DEFAULT LOW LATENCY LOW LATENCY (INTRA))
video.rateControl	CHOICE		EnumRange (CBR CAPPED-VBR)
video.level	TEXT	R/O	StringRange R/O
video.aspectRatio	CHOICE		EnumRange (AUTO 4:3 16:9)
video.entropy	CHOICE		EnumRange (CABAC CAVLC)
video.gopMode	CHOICE		EnumRange (I IP IBP IBBP IBBBP IBBBBP IBBBBBP IBBBBBBP IBBBBBBBP IB)
video.gopLength	INTEGER		IntegerRange (1 .. 255)
video.timestamps	CHOICE		EnumRange (ON OFF)
video.latencyMon	CHOICE		EnumRange (ON OFF)
video.autoAdapt	CHOICE		EnumRange (ON OFF)
video.chroma	CHOICE	R/O	EnumRange R/O (4:2:0 4:2:2)
video.bitdepth	CHOICE	R/O	EnumRange R/O (8 Bits 10 Bits)

name	*type*	*flags*	*range*
video.width	TEXT	R/O	StringRange R/O
video.height	TEXT	R/O	StringRange R/O
video.initDelay	INTEGER		IntegerRange (0 .. 5000)
video.maxDelay	INTEGER		IntegerRange (1 .. 1000)
video.pid	INTEGER		IntegerRange (0 .. 0)
video.input	TEXT	R/O	StringRange R/O
video.encryption	CHOICE		EnumRange (ON OFF)
video.aesMode	CHOICE		EnumRange (128-bit 256-bit)
video.aesKey	TEXT		StringRange
output.datarate	FLOAT		DoubleRange (0.00 .. 0.00)
output.maxrate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
output.overhead	INTEGER		IntegerRange (1 .. 500)
output.service	TEXT		StringRange
output.provider	TEXT		StringRange
output.pmtPid	INTEGER		IntegerRange (0 .. 8190)
output.pcrPid	INTEGER		IntegerRange (0 .. 8190)
output.alias	TEXT		StringRange
output.state	TEXT	R/O	StringRange R/O
audio.channels	CHOICE		EnumRange (0 1 2 3 4 5 6 7 8)
audio.1.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.1.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.1.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.1.language	TEXT	R/O	StringRange R/O
audio.1.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)

name	*type*	*flags*	*range*
audio.1.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.1.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.2.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.2.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.2.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.2.language	TEXT	R/O	StringRange R/O
audio.2.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.2.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.2.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.3.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.3.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.3.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.3.language	TEXT	R/O	StringRange R/O
audio.3.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.3.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.3.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.4.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.4.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.4.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)

name	*type*	*flags*	*range*
audio.4.language	TEXT	R/O	StringRange R/O
audio.4.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.4.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.4.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.5.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.5.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.5.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.5.language	TEXT	R/O	StringRange R/O
audio.5.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.5.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.5.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.6.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.6.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.6.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.6.language	TEXT	R/O	StringRange R/O
audio.6.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.6.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.6.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.7.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)

name	*type*	*flags*	*range*
audio.7.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.7.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.7.language	TEXT	R/O	StringRange R/O
audio.7.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.7.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.7.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
audio.8.mode	CHOICE		EnumRange (STEREO MONO LEFT MONO-RIGHT)
audio.8.datarate	CHOICE		EnumRange (16 kbps 24 kbps 32 kbps 48 kbps 56 kbps 64 kbps 96 kbps 112 kbps 128 kbps 160 kbps 192 kbps 256 kbps)
audio.8.coding	CHOICE		EnumRange (AAC-LC AAC-ELD)
audio.8.language	TEXT	R/O	StringRange R/O
audio.8.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
audio.8.samplingRate	CHOICE		EnumRange (16 kHz 32 kHz 44.1 kHz 48 kHz 96 kHz)
audio.8.input	CHOICE		EnumRange (PCM AC3 DTS AAC)
video.connector	CHOICE		EnumRange (SDI-1 SDI-2 SDI-3 SDI-4)
audio.1.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.2.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)

name	*type*	*flags*	*range*
audio.3.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.4.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.5.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.6.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.7.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
audio.8.sdiMatrix	CHOICE		EnumRange (Embedded Pair 1 Embedded Pair 2 Embedded Pair 3 Embedded Pair 4 Embedded Pair 5 Embedded Pair 6 Embedded Pair 7 Embedded Pair 8)
ipout.1.enable	CHOICE		EnumRange (ON OFF)
ipout.1.protocol	CHOICE		EnumRange (UDP RTP RTSP ZIXI PROMPEG RTMP SRT RIST)
ipout.1.interface	CHOICE		EnumRange (FRAME ETH-1 ETH-2)
ipout.1.address	TEXT		StringRange
ipout.1.port	INTEGER		IntegerRange (0 .. 65535)
ipout.1.alias	TEXT		StringRange

name	*type*	*flags*	*range*
ipout.1.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.1.udpPacketSize	CHOICE		EnumRange (188 376 564 752 940 1128 1316 1472)
ipout.1.shaping	CHOICE		EnumRange (ON OFF)
ipout.1.fecMode	CHOICE		EnumRange (1D 2D)
ipout.1.fecColumns	INTEGER		IntegerRange (0 .. 0)
ipout.1.fecRows	INTEGER		IntegerRange (0 .. 0)
zixiout.1.password	TEXT		StringRange
zixiout.1.latency	INTEGER		IntegerRange (0 .. 6000)
zixiout.1.abr	CHOICE		EnumRange (ON OFF)
zixiout.1.datarate	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
zixiout.1.maxrate	FLOAT		DoubleRange (0.00 .. 0.00)
zixiout.1.minrate	FLOAT		DoubleRange (0.00 .. 0.00)
zixiout.1.failover	CHOICE		EnumRange (ON OFF)
zixiout.1.address	TEXT		StringRange
zixiout.1.port	INTEGER		IntegerRange (0 .. 65535)
zixiout.1.fecEnable	CHOICE		EnumRange (ON OFF)
zixiout.1.overhead	INTEGER		IntegerRange (0 .. 50)
rtmpout.1.address	TEXT		StringRange
rtmpout.1.streamName	TEXT		StringRange
rtmpout.1.username	TEXT		StringRange
rtmpout.1.password	TEXT		StringRange
srtout.1.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.1.latency	INTEGER		IntegerRange (20 .. 8000)
ristout.1.profile	CHOICE		EnumRange (SIMPLE MAIN)
ristout.1.latency	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
ipout.2.enable	CHOICE		EnumRange (ON OFF)
ipout.2.protocol	CHOICE		EnumRange (UDP UDP RTP RTSP ZIXI PROMPEG RTMP SRT RIST)
ipout.2.interface	CHOICE		EnumRange (FRAME ETH-1 ETH-2)
ipout.2.address	TEXT		StringRange
ipout.2.port	INTEGER		IntegerRange (0 .. 65535)
ipout.2.alias	TEXT		StringRange
ipout.2.ttl	INTEGER		IntegerRange (1 .. 255)
ipout.2.udpPacketSize	CHOICE		EnumRange (188 376 564 752 94 1128 1316 1472)
ipout.2.shaping	CHOICE		EnumRange (ON OFF)
ipout.2.fecMode	CHOICE		EnumRange (1D 2D)
ipout.2.fecColumns	INTEGER		IntegerRange (0 .. 0)
ipout.2.fecRows	INTEGER		IntegerRange (0 .. 0)
zixiout.2.password	TEXT		StringRange
zixiout.2.latency	INTEGER		IntegerRange (0 .. 6000)
zixiout.2.abr	CHOICE		EnumRange (ON OFF)
zixiout.2.datarate	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
zixiout.2.maxrate	FLOAT		DoubleRange (0.0 .. 0.0)
zixiout.2.minrate	FLOAT		DoubleRange (0.0 .. 0.0)
zixiout.2.failover	CHOICE		EnumRange (ON OFF)
zixiout.2.address	TEXT		StringRange
zixiout.2.port	INTEGER		IntegerRange (0 .. 65535)
zixiout.2.fecEnable	CHOICE		EnumRange (ON OFF)
zixiout.2.overhead	INTEGER		IntegerRange (0 .. 50)
rtmpout.2.address	TEXT		StringRange
rtmpout.2.streamName	TEXT		StringRange

name	*type*	*flags*	*range*
rtmpout.2.username	TEXT		StringRange
rtmpout.2.password	TEXT		StringRange
srtout.2.mode	CHOICE		EnumRange (CALLER LISTENER RENDEZVOUS)
srtout.2.latency	INTEGER		IntegerRange (20 .. 8000)
ristout.2.profile	CHOICE		EnumRange (SIMPLE MAIN)
ristout.2.latency	INTEGER		IntegerRange (0 .. 0)
ipout.packetsTx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.datarate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.rtt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.latency	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.packetLoss	TEXT	R/O	StringRange R/O
ipout.packetsNonReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.peer	TEXT	R/O	StringRange R/O
ipout.packetsReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.bandwidth	TEXT	R/O	StringRange R/O
ipout.reconnects	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.packetRate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
ipout.packetsDrop	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.packetsArqReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipout.packetsFecReco	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.encoder	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Channel Status)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video input)

9.737 Vocality-FourWire

This page describes the device driver and the device window for the Vocality Four-Wire units which provides Four Wire circuits over IP for Intercom. It is a monitoring only device driver.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Overall Status
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the antenna controller.

parameter	***description***
address	The ACU usually expects the address "1".
config.remotes	The number of remotes

Network settings

Remember to configure the IP address, netmask and default gateway properly on the unit. Then use the IP address and TCP port number 23 to configure the device interface. For example: 10.0.0.123:23

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.remotes	INTEGER	SAVE SETUP	IntegerRange (0 .. 8)
config.username	TEXT	SAVE SETUP	StringRange
config.password	TEXT	SAVE SETUP	StringRange
remote.01.name	TEXT	R/O	StringRange R/O DIS
remote.01.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.01.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.01.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.name	TEXT	R/O	StringRange R/O DIS
remote.02.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.02.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.02.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.name	TEXT	R/O	StringRange R/O DIS

name	*type*	*flags*	*range*
remote.03.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.03.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.03.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.name	TEXT	R/O	StringRange R/O DIS
remote.04.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.04.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.04.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.name	TEXT	R/O	StringRange R/O DIS
remote.05.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.05.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.05.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.name	TEXT	R/O	StringRange R/O DIS
remote.06.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.06.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)

name	*type*	*flags*	*range*
remote.06.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.06.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.name	TEXT	R/O	StringRange R/O DIS
remote.07.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.07.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.07.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.name	TEXT	R/O	StringRange R/O DIS
remote.08.link	CHOICE	R/O	EnumRange R/O DIS (UP DOWN)
remote.08.rxPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.txPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.muxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.unmuxedPackets	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.seqErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
remote.08.unpackErrors	INTEGER	R/O	IntegerRange R/O DIS (0 .. 0)
internal.01	TEXT	R/O	StringRange R/O
internal.02	TEXT	R/O	StringRange R/O
internal.03	TEXT	R/O	StringRange R/O
internal.04	TEXT	R/O	StringRange R/O
internal.05	TEXT	R/O	StringRange R/O
internal.06	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.07	TEXT	R/O	StringRange R/O
internal.08	TEXT	R/O	StringRange R/O
internal.n01	TEXT	R/O	StringRange R/O
internal.n02	TEXT	R/O	StringRange R/O
internal.n03	TEXT	R/O	StringRange R/O
internal.n04	TEXT	R/O	StringRange R/O
internal.n05	TEXT	R/O	StringRange R/O
internal.n06	TEXT	R/O	StringRange R/O
internal.n07	TEXT	R/O	StringRange R/O
internal.n08	TEXT	R/O	StringRange R/O
clear	TEXT	NOPRESET	StringRange

9.738 WG-Switch

This page describes the device driver and the device window for the*WG-Switch* logical device. This device controls a latching relay providing two drive coils and two position indication circuits. In most cases this relay will be a wave-guide transfer switch, but the WG-Switch device is not limited to this application.

The *WG-Switch* logical device is part of the sat-nms framework for M&C of plain digital I/O functions. It does not interrogate a PLC device directly, but requires a low level device driver to do the physical communication. Examples for such drivers are the Fanuc-PLC9030 and the NAIS-FP0 PLC device drivers.

The relay to be controlled with the *WG-Switch* device must be a *latching* type. This means, the relay gets set to a certain position by activating one of the drive inputs with a short (e.g. 500msec) pulse. The relay moves to the commanded position and remains there without further activation until another position is commanded. Position indication contacts must be provided by the relay to make the VLC/M&C know at which position the relay actually is.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets / shows the actual switch state.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

- *parameter --- description*
- driveAPort --- The address of the PLC port which sets the switch to the "A" position. See the paragraph 'Port addressing' below for details. For a read-only switch leave this field empty
- driveBPort --- The address of the PLC port which sets the switch to the "B" position. See the paragraph 'Port addressing' below for details. For a read-only switch leave this field empty
- indicationAPort --- The address of the 'position A' indication port, which tells that the relay / wave-guide switch is in "A" position.
- indicationBPort --- The address of the 'position B' indication port, which tells that the relay / wave-guide switch is in "B" position.
- pulseWidth --- The activation pulse width (msecs) to be sent to the switch for a position change. When setting this parameter, you should be aware that pulses which are shorter than the IDLE time of the PLC driver's thread may get lost. If in doubt, set the pulse width to 1100 msecs or in the VLC interface/device setup screen tune the IDLE time of the PLC driver thread to e.g. 200 msecs. If used with the SatService IO-FEP, the pulse width must be set to 1500 msecs.
- positions --- By default the device reports the state of the output as 'A'/'B'. You may specify arbitrary texts for these states. 'positions' contains the two descriptions, separated by a comma. Please note, that the graphical "Switch" element at the sat-nms user interface *requires* the switch to use the 'A'/'B' selections.
- enableQuery --- Selects if the switch position selector at the first page of the device window shall query the operator before it actually switches. Default value is "NO".Please note, that changing this parameter only applies to device windows which are opened *after* the parameter has been changed.
- positionCheck --- Selects how the device compares the commanded switch position and the position detected by the position indication contacts:
    - ALWAYS --- The device continuously monitors the indicated switch position and compares it to the recently commanded position. A faults.02 "wrong position" fault is signalled whenever these positions differ. Any position change which has not been initiated by the WG-Switch device itself lead to a "wrong position" fault with tis

setting.
    - AFTER-SWITCHING --- The device does the position check only once, after a new position has been set by te device. If this switching failed because the switch is stuck, a "wrong position" fault is raised. Position changes from "outside" are detected and shown as the actual switch position, but the generate no fault.

- commFaultId --- By default (commFaultId is empty) the device signals a communication fault if the summary fault of the PLC device get active. This may not be exactly what you want, because the PLC device may signal a fault which does not inhibit the switching or input sampling process. You may set commFaultId to the parameter ID of a fault flag which signals the desired condition. This makes the device ignore the PLC's summary fault and use the state of this fault flag as communication fault. The SatService IO-FEP driver for example offers such a fault flag which signals that actually no switching an no input interrogation is possible (faults.17 at the IO-FEP).

Fault flags

fault flag	***fault name***	***description***
faults.99	Communication	Unless the the setup parameter commFaultId is set, faults.99 is set if the PLC device which manages the physical ports signals a fault. With commFaultId set, the fault state addressed by this parameter is replicated as communication fault of the device.
faults.01	Position Indication	This fault is set if the switch position cannot be determined unambiguously. This is if not exactly one of the indication ports is 'high'.
faults.02	Wrong Position	This fault is set if the indicated switch position does not match the commanded one. The fault remains active until either the switch position is changed or a new position is commanded.
faults.03	Configuration Setup	This fault signals that the at least one of the xxxxPort parameters listed above ist not properly configured. The device requires that both drive ports and both position indication ports are set to values which matches the syntax description shown below.

Port addressing

A device driver for a PLC gives access to it's I/O ports through a number of integer variables. Each variable stands for up to 64 input or output ports which are addressed by the bit position within the variable.

Port (bit-) addresses in the configuration of the logical I/O devices refer to such a variable which is provided by the PLC driver and also to the bit position within this variable. The syntax of a

port address specification is:

<variable-identifier> **+** *<bit-offset>*

<bit-offset> is a decimal constant in the range 0 .. 63. You may replace the **+** character in the port address by a percent sign (**%**) and specify the bit offset in hexadecimal notation. This may be helpful for PLC types which use hexadecimal addresses.

Example: The port address ' *FANUC.Q00065+1* ' refers to the *2^1* bit in the variable ' *Q00065* ' provided by the ' *FANUC* ' device. In fact this addresses the output table register '%Q00066' at the PLC. Other PLC drivers may use a different naming scheme for the variables, but the concept always is the same.

Switch positions

The *WG-Switch* logical device internally controls and monitors the switch position as A/B, following the names of the port configuration parameters *driveAPort* , *driveBPort* , *indicationAPort* and *indicationBPort* . This internal (not translated) switch position may differ from the switch position signalled with the *position* selection at the device.

Using the *positions* configuration parameter, the switch positions may be renamed to arbitrary values. E.g. the positions of a switch may be labelled "DUMMY LOAD" and "ANTENNA". You should however consider, that the graphical "Switch" element at the sat-nms user interface only works, if the selections for the switch are names "A" and "B".

Read-only mode

The switch may be configured to act as a read-only switch. The switch cannot be commanded / actuated in this mode, but it shows the actual position of the switch. For switches which are actuated by an external source, a reasonable delay time for *faults.01* (Position Indication) should be set to avoid spurious faults during the actuation if the switch.

A *WG-Switch* is configured read-only by leaving both drive port configuration parameters (*driveAPort* , *driveBPort*) empty.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.indication	TEXT	R/O	StringRange R/O
config.driveAPort	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.driveBPort	TEXT	SAVE SETUP	StringRange
config.indicationAPort	TEXT	SAVE SETUP	StringRange
config.indicationBPort	TEXT	SAVE SETUP	StringRange
config.pulseWidth	INTEGER	SAVE SETUP	IntegerRange (200 .. 5000)
config.positions	TEXT	SAVE SETUP	StringRange
config.enableQuery	CHOICE	SAVE SETUP	EnumRange (OFF ON)
config.positionCheck	CHOICE	SAVE SETUP	EnumRange (ALWAYS AFTER-SWITCHING)
config.commFaultId	TEXT	SAVE SETUP	StringRange
position	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Position Indication)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Wrong Position)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Configuration Setup)
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

9.739 WISI-Chameleon-DVBC

This page describes the *Wisi-Chamaeleon-DVBC* device driver and its device window. The driver monitors and controls one DVB-C module within a Chameleon processor unit

Multiple instances of the device may be configured in the sat-nms M&C to handle all DVB-C receivers within a Chameleon processor. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that SNMP requests to the device do not overlap.

To allow proper operation the module and input of the respective DVB-C receiver has to be selected.

The driver was tested with a single DVB-C receiver in a 1RU GN20 B/R Chassis.

- HW revision: 3001
- SW version: 3.3
- Tuner: 077 (demodulator version 60)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The rf and service parameters.
- --- The video parameters.
- --- The audio parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.
- module --- Selects the DVB-C module within a Chameleon processor.
- input --- Selects the RF input of the module the DVB-C receiver instance is using.
- deviceComment --- Define a free textual description of the unit which pop up on any screen when moving the mouse cursor other the respective device.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.name	TEXT	R/O	StringRange R/O
info.hwVersion	TEXT	R/O	StringRange R/O
info.swVersion	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.cpuUsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.memUsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.tsProcUsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.tsPidMapperUsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.tsMemUsed	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
config.module	INTEGER	SAVE SETUP	IntegerRange (1 .. 100)
config.input	CHOICE	SAVE SETUP	EnumRange (1 2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
reset	TEXT	NOPRESET	StringRange
programNo	INTEGER		IntegerRange (0 .. 0)
programList	CHOICE		EnumRange (0 - NO SERVICES)
actualProgram	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
state.level	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.ber	TEXT	R/O	StringRange R/O
state.snr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
channel	CHOICE		EnumRange (E2 E3 E4 S2 S3 S4 S5 S6 S7 S8 S9 S10 E5 E6 E7 E8 E9 E10 E11 E12 S11 S12 S13 S14 S15 S16 S17 S18 S19 S20 S21 S22 S23 S24 S25 S26 S27 S28 S29 S30 S31 S32 S33 S34 S35 S36 S37 S38 S39 S40 S41 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33 E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49 E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65 E66 E67 E68 E69 MANUAL)
symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
modulation	CHOICE		EnumRange (AUTO QAM-16 QAM-32 QAM-64 QAM-128 QAM-256)
video.convert	CHOICE		EnumRange (AUTO IGNORE LETTERBOX PANSCAN COMBINED FORCED)
video.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)
video.test	CHOICE		EnumRange (OFF ON)
video.format	TEXT	R/O	StringRange R/O
video.standard	TEXT	R/O	StringRange R/O
video.aspect	TEXT	R/O	StringRange R/O
audio.pid	INTEGER	R/O	IntegerRange R/O (0 .. 8191)

name	*type*	*flags*	*range*
audio.coding	TEXT	R/O	StringRange R/O
audio.routing	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Input signal)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Video)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Audio)
mi	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ii	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ti	INTEGER	R/O	IntegerRange R/O (0 .. 0)
di	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.740 Walton-Deice

This page describes the device driver and the device window for the Walton Deicing Controller DP-9.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The first page contains the switches to manual enable Deicing and Rain Blower and the status messages of the system.
- --- The second page contains the status of all the six heaters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Deicing Controller.

parameter	*description*
address	The address of the DP-9 or 48 for RS232.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface connected to the UNITY e.g *9600,7,E,1* regarding the configuration of the DP-9

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.device	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
config.model	CHOICE	SAVE SETUP	EnumRange (DP-8/DS-15 DP-9/DS-16)
flags.wet	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.icy	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.deice	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.feedheat	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.subheat	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.rainblow	BOOLEAN	R/O	BooleanRange R/O (true false)
db9.deice	BOOLEAN	R/O	BooleanRange R/O (true false)
db9.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.1	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.1	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
blower.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.2	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.2	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.3	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.3	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.4	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.4	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.4	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.5	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.5	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.5	BOOLEAN	R/O	BooleanRange R/O (true false)
blower.6	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.call.6	BOOLEAN	R/O	BooleanRange R/O (true false)
heater.6	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.flags.1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.5	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.6	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.7	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flags.8	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.deice	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.rain	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.manual	INTEGER		IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
deice	CHOICE		EnumRange (OFF ON)
rain	CHOICE		EnumRange (OFF ON)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Heater 1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Heater 1 overtemp)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Heater 2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Heater 2 overtemp)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Heater 3)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Heater 3 overtemp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Heater 4)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Heater 4 overtemp)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Heater 5)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Heater 5 overtemp)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Heater 6)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Heater 6 overtemp)
faults.13	ALARM	R/O	AlarmFlagRange R/O (DS-15/16 Communication)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Deice Standby)

9.741 Wavestream-BUC-SNMP

This page describes the device driver and the device window for the Wavestream block upconverter.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- *parameter --- description*
- address --- The SNMP address specification for the device. The full address specification consists of the following fields:*aaa.bbb.ccc.ddd:pppp read-community write-community*
  - *aaa.bbb.ccc.ddd* --- The IP address of the device in 'dotted quad' notation. This field is mandatory.
  - *pppp* --- The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
  - *read-community* --- The SNMP read community name. This must match the read community expected by the device.
  - *write-community* --- The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. If both community names are missing, the driver uses 'public' as a default for read and write community.

Remark

External and software tx: If the signal on Pin B is grounded to Secondary GND (Pin A), the unit will be hard- wired enabled. If the signal is left open, the unit is not hardwired enabled. Software TX Enable is logically ORd with external /TX_EN; either one will enable the unit.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.externalTx	CHOICE	R/O	EnumRange R/O (ENABLED DISABLED)
info.fanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.LED	CHOICE	SETUP	EnumRange (OFF ON)
config.savePowerupState	CHOICE	SETUP	EnumRange (NOOP SAVED SAVE NOW)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0 .. 30)
tx.attn	FLOAT		DoubleRange (0 .. 30)
meas.vdc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.idc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
internal.flthi	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltlo	HEX	R/O	HexRange R/O (0 .. 0)
internal.txon	BOOLEAN		BooleanRange (true false)
internal.txactive	BOOLEAN		BooleanRange (true false)
internal.led	BOOLEAN		BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.02	ALARM	R/O	AlarmFlagRange R/O (10MHz reference)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (HPA off)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Gate)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Inhibit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TX active)
faults.08	ALARM	R/O	AlarmFlagRange R/O (External TX enabled)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.742 Wavestream-BUC

This page describes the device driver and the device window for the Wavestream block upconverter with serial interface.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- Meter readings

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter. Default serial settings are Protocol RS485 with 9600/8N1.

parameter	*description*
address	Default address is 48 (30hex)
LED	switch alarm LED on or off
savePowerupState	save settings like attenuation

Remark

External and software tx: If the signal on Pin B is grounded to Secondary GND (Pin A), the unit will be hard- wired enabled. If the signal is left open, the unit is not hardwired enabled. Software TX Enable is logically ORd with external /TX_EN; either one will enable the unit.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.fanSpeed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.LED	CHOICE	SETUP	EnumRange (OFF ON)
config.savePowerupState	CHOICE	SETUP	EnumRange (NOOP SAVED SAVE NOW)
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.firmware	TEXT	R/O	StringRange R/O
info.voutA	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.voutB	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.gain	FLOAT		DoubleRange (0 .. 20)
tx.attn	FLOAT		DoubleRange (0 .. 20)
meas.idc	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.flthi	HEX	R/O	HexRange R/O (0 .. 0)
internal.fltlo	HEX	R/O	HexRange R/O (0 .. 0)
internal.txon	BOOLEAN		BooleanRange (true false)
internal.txactive	BOOLEAN		BooleanRange (true false)
internal.led	BOOLEAN		BooleanRange (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fan)
faults.02	ALARM	R/O	AlarmFlagRange R/O (10MHz reference)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Temperature)
faults.04	ALARM	R/O	AlarmFlagRange R/O (HPA off)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Gate)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Inhibit)
faults.07	ALARM	R/O	AlarmFlagRange R/O (TX active)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Soft TX disabled)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.743 Wegener-UNITY44XX

This page describes the device driver and the device window for the Wegener UNITY 4420/4422 IRDs. The actual version of the driver is limited to IRDs equipped with two audio output pairs.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the main receiver settings and status display.
- --- Contains the parameters concerning video and audio output.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the UNITY IRD.

parameter	***description***
loFreq1	The LO frequency to be used with the IRD input 1.
loFreq2	The LO frequency to be used with the IRD input 2.
loFreq3	The LO frequency to be used with the IRD input 3.
loFreq4	The LO frequency to be used with the IRD input 4.
ebnoThreshold	The Eb/No threshold value. The IRD mutes it's outputs if the signal quality drops below this value
ebnoMArgin	The Eb/No margin value. A "Signal quality" fault is generated if the Eb/No reading drops below the threshold+margin value.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.
Serial	Remember to set the line parameters for the serial interface connected to the UNITY. The proper setting is *19200,N,8,1* , regardless of the setting made at the front panel of the device. Always use serial port 2 of the IRD, id should be configured as "terminal" at the IRD's front panel.

Remarks

- The UNITY IRD is controlled via a terminal protocol. Specially the status reports sent by the receiver are designed to be read by a human, not by a computer. Hence the update of information ad the screen is very slow.
- The IRD shows not all program numbers of large bouquets. This is a limitation of the IRD, not of the driver.
- Audio channels are selected by their language code with the UNITY. If at the uplink the same language code is used for two channels, they cannot be assigns explicitly to the outputs of the receiver. Use the automatic selection "*" in this case.

- By default, the driver treats missing audio or video data in the MPEG stream as a fault. This may not be desirably in all cases. Switch "OFF" the fault conditions which shall not be monitored at the faults page. This does not influence the status display at the primary parameter page.

Low Level Configuration Commands

When using a UNITY IRD the first time, the device should be initialized with the commands shown below. They may be entered through the "low level" command facility of the software or by means of a RS232 terminal connected to the UNITY's terminal port.

- **reset** <ENTER> **bacdore** <ENTER> (or use second Password) **vidtest** <ENTER> **sethdelayp 3e** <ENTER> **sethdelayn 34** <ENTER> **eewrite h ed 0** <ENTER> **isog e** <ENTER> **bacdore** <ENTER> **reset** <ENTER> --- Reset Unit Password Vidtest PAL Video Horizontal Delay NTSC Video Horizontal Delay COMPEL Disabled Bypass TAG Password Reset Unit

Password Setting (Change Password)

- **reset** <ENTER> **tecaligi** <ENTER> **vistest** <ENTER> **eewrite h 131 ff** <ENTER> **reset** <ENTER> --- Reset Unit Second Password Vidtest Changed tecaligi to bacdore Reset Unit

Trouble Shooting

The list below shows some troubleshooting hints for the UNITY IRD, given by Wegener. Not all items apply to monitoring & control in particular, they however may be helpful during system integration.

- Unit will not power up and/ or front panel does not work. ---
    - Verify that unit is connected to a good AC source.
    - Verify that all wires and cables are connected inside of the unit.
    - Check to see if the unit will respond to terminal commands and give the desired outputs.
    - If the unit does, try replacing the front panel from a unit that has bad outputs if available.
    - If none of the above items work, contact the factory for further instructions.
- No audio out. ---
    - Confirm that the app code version is 1.2 or later.
    - Confirm that a valid audio designator (program) has been selected.
    - Verify that the external connections and equipment is working properly.
- No AES/EBU output. ---
    - Confirm that the analog audio outputs are working.
    - Confirm that the ribbon cable from the AES/EBU port is connected properly to J12 on the main motherboard.
- No composite video out. ---

- Confirm that the app code is 1.2 or later.
- Confirm that the receiver is actually seeing program information via the front panel or terminal command "r m". If no programs is available, verify that the transmit equipment is working properly and reset the receiver. If programs are available, confirm that the receiver is set to a program with active video.
- Check the output connector and confirm that it has not been broken.

- Composite video out, but with problems. ---
  - If the video is not centered on the screen as it should be, confirm tha the Pal horizontal delay is set to "3e". Use the terminal commands: **bacdore** <enter> **sethdelayp** <enter>. If it is not "3e", then type:**sethdelayp 3e** <enter> (PAL only)
  - Incorrect levels - confirm that proper terminations are being used
  - Adjust R52 only if the level is out of spec. BE EXTREMELY CAREFUL WHILE ADJUSTING THIS PART. IT IS A SURFACE MOUNT POT. R52 is located at the rear of the unit.

- No SDI output or bad SDI output. ---
  - Confirm that the composite video output is working properly.
  - Confirm that the SDI output connector is not broken.
  - Confirm that the SDI parts are programmed with the correct version of code(serial5.jed) for PAL. The only way you can confirm the version of code is to use the JTAG programmer and verify the programming of the Xilinx part.

- Audio/video sync is off. ---
  - Confirm that the A/VSync is set to 0"(zero) via the terminal by using the following commands: **bacdore** <enter> **vidtest** <enter> **capcr 0** <enter>.

- The receiver will not lock to the carrier. ---
  - Verify that you have a good signal.
  - Verify that the LNB lo's are set correctly.
  - Verify that the receiver says "bypassing tags" at the bottom of the "r c" report. If it does not, type: **isog e** <enter>.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.loFreq1	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.loFreq2	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.loFreq3	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.loFreq4	FLOAT	SETUP	DoubleRange (0.00 .. 0.00)
config.ebnoThreshold	FLOAT	SETUP	DoubleRange (2.0 .. 7.0)
config.ebnoMargin	FLOAT	SETUP	DoubleRange (1.0 .. 11.9)
config.deviceComment	TEXT	SAVE SETUP	StringRange
internal.a1.mode	TEXT	R/O	StringRange R/O
internal.a1.mute	TEXT	R/O	StringRange R/O
internal.a1.star	TEXT	R/O	StringRange R/O
internal.a2.mode	TEXT	R/O	StringRange R/O
internal.a2.mute	TEXT	R/O	StringRange R/O
internal.a2.star	TEXT	R/O	StringRange R/O
internal.drate	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
info.serial	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.application	TEXT	R/O	StringRange R/O
info.receiver	TEXT	R/O	StringRange R/O
info.video-dec	TEXT	R/O	StringRange R/O
info.audio-dec	TEXT	R/O	StringRange R/O
info.descrambler	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.boot	TEXT	R/O	StringRange R/O
info.audio-demux	TEXT	R/O	StringRange R/O
info.encryption	TEXT	R/O	StringRange R/O
info.lattice	TEXT	R/O	StringRange R/O
info.cond-access	TEXT	R/O	StringRange R/O
info.audioPorts	INTEGER	R/O	IntegerRange R/O (0 .. 0)
input	CHOICE		EnumRange (1 2 3 4)
frequency	FLOAT		DoubleRange (0.00 .. 0.00)
dataRate	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	FLOAT		DoubleRange (0.000 .. 0.000)
fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 7/8)
modulation	CHOICE		EnumRange (QPSK)
programNo	TEXT		StringRange
programList	TEXT	R/O	StringRange R/O
audioList	TEXT	R/O	StringRange R/O
actualProgram	TEXT	R/O	StringRange R/O
video.format	CHOICE		EnumRange (AUTO)
video.aspect	CHOICE		EnumRange (AUTO)
video.output	CHOICE		EnumRange (OFF NRM)
video.test	CHOICE		EnumRange (TST NRM)
audio.1.program	TEXT		StringRange
audio.1.output	CHOICE		EnumRange (NRM LFT RGT REV OFF)
audio.1.level	INTEGER		IntegerRange (0 .. 14)
audio.1.test	CHOICE		EnumRange (NRM)
audio.1.info	TEXT	R/O	StringRange R/O
audio.2.program	TEXT		StringRange

name	*type*	*flags*	*range*
audio.2.output	CHOICE		EnumRange (NRM LFT RGT REV OFF)
audio.2.level	INTEGER		IntegerRange (0 .. 14)
audio.2.test	CHOICE		EnumRange (NRM)
audio.2.info	TEXT	R/O	StringRange R/O
flags.lock	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mpeg	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.video	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.audio	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.ebno	BOOLEAN	R/O	BooleanRange R/O (true false)
state.ebno	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.vMode	TEXT	R/O	StringRange R/O
state.vAspect	TEXT	R/O	StringRange R/O
reset	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Carrier lock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (MPEG lock)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Video lock)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Audio lock)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Signal quality)

9.744 Wilmers-blueberryCOMPACT

This page describes the device driver and the device window for the*Wilmers blueberryCOMPACT (aka NDL485 compact)* weather data logger. The sat-nms device driver for the blueberryCOMPACT reads the numeric measurement data from the device's Web GUI (left frame).

The driver does not provide any means to configure or setup the device. This has to be done using the device's Web GUI.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the actual measurement values.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window the device address must be configured in order to make the sat-nms software talk to the device.

parameter	***description***
address	As the device uses HTTP basic authentication to protect it's Web GUI, user name ('guest') and password have to be specified together with the device's IP address in this field. Example: **10.0.0.1 guest mypasswd**

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
radiation	FLOAT	R/O	DoubleRange R/O (0.0000 .. 0.0000)
windDirection	FLOAT	R/O	DoubleRange R/O (0 .. 0)
windRose	TEXT	R/O	StringRange R/O
windSpeed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
verticalWindSpeed	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
airPressure	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
airHumidity	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
airTemp	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
dewPoint	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rain	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rainToday	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
rainMonth	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
supplyLogger	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
supplyMetPak	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery1	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
battery2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
fog	BOOLEAN	R/O	BooleanRange R/O (true false)
snow	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Fog)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Snow)

9.745 Work-AT61-Modulator

This page describes the device driver and the device window for the Work Microwave DVB-S/S2/S2X Modulator AT-61 via SNMP.

This driver includes Carried ID parameter.

It was tested with:

- *Modulator version: BBA01-51/MOK03.26
- Linux version: AOC02.11.12
- Busbridge version: ACB02.05

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The modulator settings.

- --- The Carrier ID settings.
- --- The ASI and TSoIP input settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	***description***
address	The SNMP address specification for the device (details below).
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
banner	Front panel banner
lowLevel	Defines the lower limit of the modulators output level
highLevel	Defines the upper limit of the modulators output level

SNMP address specification

The full address consists of the following fields:
`aaa.bbb.ccc.ddd:pppp read-community write-community`

parameter	**description**
aaa.bbb.ccc.ddd	The IP address of the device in 'dotted quad' notation. This field is mandatory.
pppp	The IP port number where the device accepts SNMP requests. The port number may be omitted, the driver uses the standard port 161 in this case.
read-community	The SNMP read community name. This must match the read community expected by the device.
write-community	The SNMP write community name. This must match the write community expected by the device. The write community name may be omitted, the driver uses the read community name for both, read and write in this case. The device uses `public` as a default for read and `private` as write community string.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.modulatorVersion	TEXT	R/O	StringRange R/O
info.busbridgeVersion	TEXT	R/O	StringRange R/O
info.linuxVersion	TEXT	R/O	StringRange R/O
info.bootloaderVersion	TEXT	R/O	StringRange R/O
info.modulatorFPGAVersion	TEXT	R/O	StringRange R/O
info.temp.device	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.bridgeBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.bridgeChip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.cpuBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.cpuChip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.banner	TEXT	SETUP	StringRange
config.lowLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
config.highLevel	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.lowFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highFreqLim	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.lowSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.highSymRateLim	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.rollOff	CHOICE		EnumRange (20% 25% 35% 5% 10% 15%)
tx.power	FLOAT		DoubleRange (-30.0 .. 0.0)
tx.mod.refOutput	CHOICE		EnumRange (OFF ON)
tx.mod.slopeComp	INTEGER		IntegerRange (-9 .. 9)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.cwMode	CHOICE		EnumRange (OFF CW)
tx.mod.processing	CHOICE		EnumRange (OFF ENCAPSULATION STREAMING)
tx.mod.plScrambling	INTEGER		IntegerRange (0 .. 262141)
tx.mod.multistream	CHOICE		EnumRange (OFF ON TIMESLICING)

name	*type*	*flags*	*range*
tx.mod.ccmLegacyMode	CHOICE		EnumRange (OFF 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63)
			EnumRange (UNKNOWN DVB-S2-QPSK-1/4-NORMAL DVB-S2-QPSK-1/4-SHORT DVB-S2-QPSK-1/3-NORMAL DVB-S2-QPSK-1/3-SHORT DVB-S2-QPSK-2/5-NORMAL DVB-S2-QPSK-2/5-SHORT DVB-S2-QPSK-1/2-NORMAL DVB-S2-QPSK-1/2-SHORT DVB-S2-QPSK-3/5-NORMAL DVB-S2-QPSK-3/5-SHORT DVB-S2-QPSK-2/3-NORMAL DVB-S2-QPSK-2/3-SHORT DVB-S2-QPSK-3/4-NORMAL DVB-S2-QPSK-3/4-SHORT DVB-S2-QPSK-4/5-NORMAL DVB-S2-QPSK-4/5-SHORT DVB-S2-QPSK-5/6-NORMAL DVB-S2-QPSK-5/6-SHORT DVB-S2-QPSK-8/9-NORMAL DVB-S2-QPSK-8/9-SHORT DVB-S2-QPSK-9/10-NORMAL DVB-S2-8PSK-3/5-NORMAL DVB-S2-8PSK-3/5-SHORT DVB-S2-8PSK-2/3-NORMAL DVB-S2-8PSK-2/3-SHORT DVB-S2-8PSK-3/4-NORMAL DVB-S2-8PSK-3/4-SHORT DVB-S2-8PSK-5/6-NORMAL DVB-S2-8PSK-5/6-SHORT DVB-S2-8PSK-8/9-NORMAL DVB-S2-8PSK-8/9-SHORT DVB-S2-8PSK-9/10-NORMAL DVB-S2-16APSK-2/3-NORMAL DVB-S2-16APSK-2/3-SHORT DVB-S2-16APSK-3/4-NORMAL DVB-S2-16APSK-3/4-SHORT DVB-S2-16APSK-4/5-

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		NORMAL DVB-S2-16APSK-4/5- SHORT DVB-S2-16APSK-5/6- NORMAL DVB-S2-16APSK-5/6- SHORT DVB-S2-16APSK-8/9- NORMAL DVB-S2-16APSK-8/9- SHORT DVB-S2-16APSK-9/10- NORMAL DVB-S2-32APSK-3/4- NORMAL DVB-S2-32APSK-3/4- SHORT DVB-S2-32APSK-4/5- NORMAL DVB-S2-32APSK-4/5- SHORT DVB-S2-32APSK-5/6- NORMAL DVB-S2-32APSK-5/6- SHORT DVB-S2-32APSK-8/9- NORMAL DVB-S2-32APSK-8/9- SHORT DVB-S2-32APSK-9/10- NORMAL DVB-S2X-QPSK-13/45- NORMAL DVB-S2X-QPSK-9/20- NORMAL DVB-S2X-QPSK-11/20- NORMAL6 DVB-S2X-8PSK-5/9- LINEAR DVB-S2X-8PSK-26/45- LINEAR DVB-S2X-8PSK-23/36- NORMAL DVB-S2X-8PSK-25/36- NORMAL DVB-S2X-8PSK-13/18- NORMAL DVB-S2X-16APSK-1/2- LINEAR DVB-S2X-16APSK-8/15- LINEAR DVB-S2X-16APSK-5/9- LINEAR DVB-S2X-16APSK-26/45- NORMAL DVB-S2X-16APSK-3/5- NORMAL DVB-S2X-16APSK-3/5- LINEAR DVB-S2X-16APSK-28/45- NORMAL DVB-S2X-16APSK-23/36- NORMAL DVB-S2X-16APSK-2/3- LINEAR DVB-S2X-16APSK-25/36- NORMAL DVB-S2X-16APSK-13/18- NORMAL DVB-S2X-16APSK-7/9- NORMAL DVB-S2X-16APSK-77/90- NORMAL DVB-S2X-32APSK-2/3- LINEAR DVB-S2X-32APSK-32/45- NORMAL DVB-S2X-32APSK-11/15- NORMAL DVB-S2X-32APSK-7/9- NORMAL DVB-S2X-64APSK-32/45- LINEAR DVB-S2X-64APSK-11/15- NORMAL DVB-S2X-64APSK-7/9- NORMAL DVB-S2X-64APSK-4/5- NORMAL DVB-S2X-64APSK-5/6- NORMAL DVB-S2X-128APSK-3/4- NORMAL DVB-S2X-128APSK-7/9-

name	*type*	*flags*	*range*
			NORMAL DVB-S2X-256APSK-29/45-LINEAR DVB-S2X-256APSK-2/3-LINEAR DVB-S2X-256APSK-31/45-LINEAR DVB-S2X-256APSK-32/45-NORMAL DVB-S2X-256APSK-11/15-LINEAR DVB-S2X-256APSK-3/4-NORMAL DVB-S2X-QPSK-11/45-SHORT DVB-S2X-QPSK-4/15-SHORT DVB-S2X-QPSK-14/45-SHORT DVB-S2X-QPSK-7/15-SHORT DVB-S2X-QPSK-8/15-SHORT DVB-S2X-QPSK-32/45-SHORT DVB-S2X-8PSK-7/15-SHORT DVB-S2X-8PSK-8/15-SHORT DVB-S2X-8PSK-26/45-SHORT DVB-S2X-8PSK-32/45-SHORT DVB-S2X-16APSK-7/15-SHORT DVB-S2X-16APSK-8/15-SHORT DVB-S2X-16APSK-26/45-SHORT DVB-S2X-16APSK-3/5-SHORT DVB-S2X-16APSK-32/45-SHORT DVB-S2X-32APSK-2/3-SHORT DVB-S2X-32APSK-32/45-SHORT DVB-S-QPSK-1/2 DVB-S-QPSK-2/3 DVB-S-QPSK-3/4 DVB-S-QPSK-5/6 DVB-S-QPSK-6/7 DVB-S-QPSK-7/8)
tx.mod.standard	CHOICE		EnumRange (DVB-S DVB-S2 DVB-S2X UNKNOWN)
tx.mod.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A UNKNOWN)
tx.mod.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 UNKNOWN)
tx.mod.frameSize	CHOICE		EnumRange (NORMAL SHORT LINEAR N/A)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.stillPicture	CHOICE		EnumRange (OFF ON)
tx.pmtPid	INTEGER		IntegerRange (16 .. 8190)
tx.videoPid	INTEGER		IntegerRange (16 .. 8190)
input.1	CHOICE		EnumRange (IP ASI-1)
input.2	CHOICE		EnumRange (IP ASI-2)
ipin.activeIfc1	CHOICE		EnumRange (A B AUTO SEAMLESS)
ipin.activeIfc2	CHOICE		EnumRange (C D AUTO SEAMLESS)
ipin.active	CHOICE		EnumRange (FROM ACTIVE 1 FROM ACTIVE 2 AUTO SEAMLESS)
ipin.packetSize	CHOICE		EnumRange (188 204)
ipin.faultTime	INTEGER		IntegerRange (0 .. 0)
ipin.1.protocol	CHOICE		EnumRange (OFF UDP RTP)
ipin.1.port	INTEGER		IntegerRange (0 .. 65535)
ipin.1.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange
ipin.1.delay	INTEGER		IntegerRange (0 .. 0)
ipin.1.inputSel	CHOICE		EnumRange (A B C D NONE)
ipin.2.protocol	CHOICE		EnumRange (OFF UDP RTP)
ipin.2.port	INTEGER		IntegerRange (0 .. 65535)
ipin.2.mode	CHOICE		EnumRange (MULTICAST UNICAST)

name	*type*	*flags*	*range*
ipin.2.address	TEXT		StringRange
ipin.2.delay	INTEGER		IntegerRange (0 .. 0)
ipin.2.inputSel	CHOICE		EnumRange (A B C D NONE)
ipin.3.protocol	CHOICE		EnumRange (OFF UDP RTP)
ipin.3.port	INTEGER		IntegerRange (0 .. 65535)
ipin.3.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.3.address	TEXT		StringRange
ipin.3.delay	INTEGER		IntegerRange (0 .. 0)
ipin.3.inputSel	CHOICE		EnumRange (A B C D NONE)
ipin.4.protocol	CHOICE		EnumRange (OFF UDP RTP)
ipin.4.port	INTEGER		IntegerRange (0 .. 65535)
ipin.4.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.4.address	TEXT		StringRange
ipin.4.delay	INTEGER		IntegerRange (0 .. 0)
ipin.4.inputSel	CHOICE		EnumRange (A B C D NONE)
tx.mux.cid.enable	CHOICE	NOPRESET	EnumRange (OFF ON)
tx.mux.cid.carrierId	TEXT	R/O	StringRange R/O
tx.mux.cid.telnNo	TEXT	NOPRESET	StringRange
tx.mux.cid.longitude	FLOAT	NOPRESET	DoubleRange (-180.000000 .. 180.000000)
tx.mux.cid.userInfo	TEXT	NOPRESET	StringRange
tx.mux.cid.latitude	FLOAT	NOPRESET	DoubleRange (-90.000000 .. 90.000000)
tx.mod.bbFrame	CHOICE		EnumRange (OFF MIN-ON MAX-ON MIN-MAX)
resetWarnings	TEXT	NOPRESET	StringRange
resetFaults	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
internal.tx.on	TEXT	R/O	StringRange R/O
internal.type	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Global)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Global Warning)

9.746 Work-Amplifier

This page describes the device driver and the device window for the Work Amplifier series

The driver was tested with the following models:

- L-Band Amplifier 950-2150Mhz

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).

parameter	*description*
autoSave	Store current parameters automatically in the instrument startup configuration.

Remote Control

- Default TCP Port = 50505
- Default Address = A

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.AmpModuleVersion	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
on	CHOICE		EnumRange (OFF ON)
gain	FLOAT		DoubleRange (0.0 .. 20.0)

name	*type*	*flags*	*range*
attn	FLOAT		DoubleRange (0.0 .. 20.0)
eq	FLOAT		DoubleRange (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Amplifier module)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal module)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Module comm.)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Ext. Mute)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Temp. Sensor)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Key)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Real time clock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Display)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Ethernet)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.12	ALARM	R/O	AlarmFlagRange R/O (High temperature)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Low temperature)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Webserver)

name	*type*	*flags*	*range*
faults.15	ALARM	R/O	AlarmFlagRange R/O (Webcontrol)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Message bus)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RF Off)

9.747 Work-DualSBD

This page describes the device driver and the device window for the Work Microwave VSCD/VHCD downconverter series with two channels

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters.
- --- The meter readings.
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.

parameter	*description*
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port = 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 20.0)
rx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.gain1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gain2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.748 Work-DualSBU

This page describes the device driver and the device window for the Work Microwave Synthesized dual block upconverters VSBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
accessMode	**Full Control:** Enables changes on all tx parameters. **Blocked by RSCM:** Disables changes on all tx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.gainRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.eqRange	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.hasFallback	CHOICE	SAVE SETUP	EnumRange (NO YES)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain.1	FLOAT		DoubleRange (0.0 .. 20.0)
tx.gain.2	FLOAT		DoubleRange (0.0 .. 20.0)

name	*type*	*flags*	*range*
tx.attn.1	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 20.0)
tx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gain1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.gain2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.749 Work-DualSSBD

This page describes the device driver and the device window for the Work Microwave Synthesized dual block downconverters SSBD.

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters
- --- The meter readings
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.loFreq.2	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)

name	*type*	*flags*	*range*
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 20.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 20.0)
rx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.750 Work-DualSSBU

This page describes the device driver and the device window for the Work Microwave

Synthesized dual block upconverters SSBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
accessMode	**Full Control:** Enables changes on all tx parameters**Blocked by RSCM:** Disables changes on all tx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.loFreq.2	FLOAT	SETUP	DoubleRange (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain.1	FLOAT		DoubleRange (0.0 .. 20.0)
tx.gain.2	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 20.0)
tx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.751 Work-DualVSCD

This page describes the device driver and the device window for the Work Microwave VSCD/VHCD downconverter series with one or two channels.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **L4:** 1600 - 1800MHz, **S:** 2000 - 2300MHz, **C:** 3400 - 4200MHz, **C1:** 3400 - 4800MHz, **X:** 7250 - 7750MHz, **KU:** 10700 - 12750MHz, **KU4:** 9300 - 9500MHz, **KA:** 18100 - 21200MHz, **KA1:** 31800 - 32300MHz, **KA2:** 17700 - 19500MHz, **KA4:** 27500 - 31000MHz, **KA15-720:** 24000 - 27000MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.
dualband	Defines if the unit is single or dual band.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L2 L4 S S1 S4 C C1 X X3 KU KU2 KU4 KA KA1 KA1b KA2 KA3 KA4 KA7 KA21 KA15-720)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
config.dualBand	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.singleChanMute	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (OFF ON)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.on.2	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.frequency.1	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.frequency.2	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.eq.1	INTEGER		IntegerRange (-64 .. 64)
rx.eq.2	INTEGER		IntegerRange (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.752 Work-DualVSCU

This page describes the device driver and the device window for the Work Microwave VSCU/VHCU upconverter series with one or two channels.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **S:** 2000 - 2300MHz, **C:** 5850 - 6650MHz, **C1:** 5850 - 7200MHz, **X:** 7900 - 8400MHz, **X3:** 7000 - 9000MHz, **KU:** 12750 - 14500MHz, **K:** 17300 - 18400MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.
dualband	Defines if the unit is single or dual band.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L2 S S4 C C1 X X3 X4 X6 KU KU1 KU4 K KA KA1 KA2 KA3 KA5 KA21)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.dualBand	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)

name	*type*	*flags*	*range*
tx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency.1	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
tx.frequency.2	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
tx.eq.1	INTEGER		IntegerRange (-64 .. 64)
tx.eq.2	INTEGER		IntegerRange (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.753 Work-MBU-MiniBuc

This page describes the device driver and the device window for the Work Satcom MiniBUC block upconverter, a rack-mount unit without front panel interface.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	*description*
address	Device device's multipoint communication address. This is a single upper case character ('A', 'B', ...) which must match the setting made at the device's front panel.
extMuteInput	Enables or disables the external mute input of the device.
powerDisplayMode	Setting this to ATTN show the current attenuator settings to the user GAIN the calculated gain (reverse of attenuation) Do not mix the to operating modes
Serial	Please refer to the Work user manual about changing interface settings. Settings used in the sat-nms software must match the serial interface parameter of the device. A common setting is to use *19200,N,8,1* .

Network Communication

This device driver may be used to control the Work Satcom modulator/upconverter through a serial RS422 interface or via a network connection. To enable the latter mode, set the network

interface parameters of the device at it's front panel to values which are suitable to your network layout. At the device setup window , set the device's IP address / IP port instead of a tty interface name.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.extRefDetected	BOOLEAN	R/O	BooleanRange R/O (true false)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 20.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 20.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.faults	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (MW-PLL unlocked)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (MW-PLL Tune Voltage limit)
faults.03	ALARM	R/O	AlarmFlagRange R/O (OCXO Oven cold)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Ref-PLL unlocked)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.754 Work-QuadSBD

This page describes the device driver and the device window for the Work Microwave synthesised quad/4-channel block downconverters VSBD.

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters
- --- The meter readings
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.

parameter	*description*
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
loFreq.3	Defines the LO Frequency for Band 3.
loFreq.4	Defines the LO Frequency for Band 4.
accessMode	**Full Control:** Enables changes on all rx parameters. **Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.4	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.3	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.4	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.3	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.4	FLOAT		DoubleRange (0.0 .. 30.0)

name	*type*	*flags*	*range*
rx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.3	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.4	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.755 Work-QuadSBU

This page describes the device driver and the device window for the Work Microwave Synthesised quad / 4-channel block upconverters VSBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters

- --- The meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
loFreq.3	Defines the LO Frequency for Band 3.
loFreq.4	Defines the LO Frequency for Band 4.
accessMode	**Full Control:** Enables changes on all tx parameters**Blocked by RSCM:** Disables changes on all tx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.4	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)

name	*type*	*flags*	*range*
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.3	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.4	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.3	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.4	FLOAT		DoubleRange (0.0 .. 30.0)
tx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.3	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.4	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)

name	*type*	*flags*	*range*
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.756 Work-RSC

This page describes the device driver and the device window for the Work Microwave RSC 1:1 Redundancy Switch

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

- address --- The device address (decimal) as set at the device's front panel.
- *parameter --- description*
- readBackDelay --- Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
- operationMode --- Choose units redundancy operation mode.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.operationMode	CHOICE	SETUP	EnumRange (one time auto reactivation priority A priority B)
protection	CHOICE		EnumRange (OFF ON TEMP-OFF)
recovDelay	INTEGER		IntegerRange (0 .. 65535)
location	CHOICE		EnumRange (1 2)
position	CHOICE		EnumRange (A B)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.03	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Relays 1)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Relays 2)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Indicator supply shortage)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.11	ALARM	R/O	AlarmFlagRange R/O (UNIT A)
faults.12	ALARM	R/O	AlarmFlagRange R/O (UNIT B)

9.757 Work-RSCM-Dual

This page describes the device driver and the device window for the Work Microwave RSCM N:1 Redundancy Switch when used with 2-channel up/downconverters.

- Works-RSCM should be used for single channel up or downconverters
- Works-RSCM-Dual should be used for dual channel up or downconverters
- Works-RSCM-Triple should be used for triple channel up or downconverters
- Works-RSCM-Quad should be used for quad channel up or downconverters

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch parameters
- --- The channel parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The device address (decimal) as set at the device's front panel.
- converterType --- The units converter type. Informational only.
- extRef.x --- The external reference source of the individual unit.
  - INTERNAL
  - EXTERNAL
  - AUTO
- bandSelect.x --- The band selection of the individual unit.
  - BAND-1
  - BAND-2

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.revision1	TEXT	R/O	StringRange R/O
info.revision2	TEXT	R/O	StringRange R/O
config.converterType	CHOICE	SAVE SETUP	EnumRange (UNKNOWN UPCONVERTER DOWNCONVERTER)
config.extRef.0	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.1	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.2	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.3	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.4	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.5	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.6	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.7	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.8	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.bandSelect.0	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.1	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.2	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.3	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.4	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.5	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.6	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.7	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)

name	*type*	*flags*	*range*
config.bandSelect.8	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
internal.0	TEXT	R/O	StringRange R/O
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O
internal.4	TEXT	R/O	StringRange R/O
internal.5	TEXT	R/O	StringRange R/O
internal.6	TEXT	R/O	StringRange R/O
internal.7	TEXT	R/O	StringRange R/O
internal.8	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8 OFF)
location.1	CHOICE	R/O	EnumRange R/O (A B)
location.2	CHOICE	R/O	EnumRange R/O (A B)
location.3	CHOICE	R/O	EnumRange R/O (A B)
location.4	CHOICE	R/O	EnumRange R/O (A B)
location.5	CHOICE	R/O	EnumRange R/O (A B)
location.6	CHOICE	R/O	EnumRange R/O (A B)
location.7	CHOICE	R/O	EnumRange R/O (A B)
location.8	CHOICE	R/O	EnumRange R/O (A B)
chain.0.enable	CHOICE		EnumRange DIS (OFF ON)
chain.1.enable	CHOICE		EnumRange DIS (OFF ON)
chain.2.enable	CHOICE		EnumRange DIS (OFF ON)
chain.3.enable	CHOICE		EnumRange DIS (OFF ON)
chain.4.enable	CHOICE		EnumRange DIS (OFF ON)
chain.5.enable	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
chain.6.enable	CHOICE		EnumRange DIS (OFF ON)
chain.7.enable	CHOICE		EnumRange DIS (OFF ON)
chain.8.enable	CHOICE		EnumRange DIS (OFF ON)
chain.0.name	TEXT	R/O	StringRange R/O DIS
chain.1.name	TEXT	R/O	StringRange R/O DIS
chain.2.name	TEXT	R/O	StringRange R/O DIS
chain.3.name	TEXT	R/O	StringRange R/O DIS
chain.4.name	TEXT	R/O	StringRange R/O DIS
chain.5.name	TEXT	R/O	StringRange R/O DIS
chain.6.name	TEXT	R/O	StringRange R/O DIS
chain.7.name	TEXT	R/O	StringRange R/O DIS
chain.8.name	TEXT	R/O	StringRange R/O DIS
chain.0.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
ch0.on	CHOICE		EnumRange DIS (OFF ON)
ch0.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch0.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch0.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.on	CHOICE		EnumRange DIS (OFF ON)
ch1.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch1.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch1.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.on	CHOICE		EnumRange DIS (OFF ON)
ch2.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch2.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch2.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.on	CHOICE		EnumRange DIS (OFF ON)
ch3.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch3.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch3.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch3.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.on	CHOICE		EnumRange DIS (OFF ON)
ch4.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch4.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch4.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.on	CHOICE		EnumRange DIS (OFF ON)
ch5.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch5.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch5.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch5.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.on	CHOICE		EnumRange DIS (OFF ON)
ch6.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch6.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch6.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.on	CHOICE		EnumRange DIS (OFF ON)
ch7.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch7.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch7.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.on	CHOICE		EnumRange DIS (OFF ON)
ch8.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch8.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch8.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch8.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
internal.l	CHOICE	R/O	EnumRange R/O (OFF AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fault)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fault)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Fault)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Fault)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Fault)

9.758 Work-RSCM-Quad

This page describes the device driver and the device window for the Work Microwave RSCM N:1 Redundancy Switch when used with 4-channel up/downconverters.

- Works-RSCM should be used for single channel up or downconverters
- Works-RSCM-Dual should be used for dual channel up or downconverters
- Works-RSCM-Triple should be used for triple channel up or downconverters

- Works-RSCM-Quad should be used for quad channel up or downconverters

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch parameters
- --- The channel parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The device address (decimal) as set at the device's front panel.
- converterType --- The units converter type. Informational only.
- extRef.x --- The external reference source of the individual unit.
  - INTERNAL
  - EXTERNAL
  - AUTO
- bandSelect.x --- The band selection of the individual unit.
  - BAND-1
  - BAND-2
- hasEqualizer --- Defines if the units are equipped with an equalizer.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.serial	TEXT	R/O	StringRange R/O
info.revision1	TEXT	R/O	StringRange R/O
info.revision2	TEXT	R/O	StringRange R/O
config.converterType	CHOICE	SAVE SETUP	EnumRange (UNKNOWN UPCONVERTER DOWNCONVERTER)
config.extRef.0	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.1	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.2	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.3	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.4	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.5	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.6	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.7	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.8	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.bandSelect.0	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.1	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.2	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.3	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.4	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.5	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.6	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.7	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)

name	*type*	*flags*	*range*
config.bandSelect.8	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
internal.0	TEXT	R/O	StringRange R/O
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O
internal.4	TEXT	R/O	StringRange R/O
internal.5	TEXT	R/O	StringRange R/O
internal.6	TEXT	R/O	StringRange R/O
internal.7	TEXT	R/O	StringRange R/O
internal.8	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8 OFF)
location.1	CHOICE	R/O	EnumRange R/O (A B)
location.2	CHOICE	R/O	EnumRange R/O (A B)
location.3	CHOICE	R/O	EnumRange R/O (A B)
location.4	CHOICE	R/O	EnumRange R/O (A B)
location.5	CHOICE	R/O	EnumRange R/O (A B)
location.6	CHOICE	R/O	EnumRange R/O (A B)
location.7	CHOICE	R/O	EnumRange R/O (A B)
location.8	CHOICE	R/O	EnumRange R/O (A B)
chain.0.enable	CHOICE		EnumRange DIS (OFF ON)
chain.1.enable	CHOICE		EnumRange DIS (OFF ON)
chain.2.enable	CHOICE		EnumRange DIS (OFF ON)
chain.3.enable	CHOICE		EnumRange DIS (OFF ON)
chain.4.enable	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
chain.5.enable	CHOICE		EnumRange DIS (OFF ON)
chain.6.enable	CHOICE		EnumRange DIS (OFF ON)
chain.7.enable	CHOICE		EnumRange DIS (OFF ON)
chain.8.enable	CHOICE		EnumRange DIS (OFF ON)
chain.0.name	TEXT	R/O	StringRange R/O DIS
chain.1.name	TEXT	R/O	StringRange R/O DIS
chain.2.name	TEXT	R/O	StringRange R/O DIS
chain.3.name	TEXT	R/O	StringRange R/O DIS
chain.4.name	TEXT	R/O	StringRange R/O DIS
chain.5.name	TEXT	R/O	StringRange R/O DIS
chain.6.name	TEXT	R/O	StringRange R/O DIS
chain.7.name	TEXT	R/O	StringRange R/O DIS
chain.8.name	TEXT	R/O	StringRange R/O DIS
chain.0.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
ch0.on	CHOICE		EnumRange DIS (OFF ON)
ch0.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch0.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch0.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch0.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch0.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch0.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.on	CHOICE		EnumRange DIS (OFF ON)
ch1.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch1.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch1.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.on	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
ch2.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch2.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch2.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.on	CHOICE		EnumRange DIS (OFF ON)
ch3.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch3.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch3.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch3.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.on	CHOICE		EnumRange DIS (OFF ON)
ch4.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch4.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch4.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.on	CHOICE		EnumRange DIS (OFF ON)
ch5.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch5.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch5.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch5.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.on	CHOICE		EnumRange DIS (OFF ON)
ch6.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch6.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch6.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch6.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.on	CHOICE		EnumRange DIS (OFF ON)
ch7.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch7.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch7.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.on	CHOICE		EnumRange DIS (OFF ON)
ch8.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch8.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch8.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch8.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.d	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.d	INTEGER		IntegerRange DIS (-64 .. 64)
internal.l	CHOICE	R/O	EnumRange R/O (OFF AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fault)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fault)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Fault)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Fault)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Fault)

9.759 Work-RSCM-Triple

This page describes the device driver and the device window for the Work Microwave RSCM N:1

Redundancy Switch when used with 3-channel up/downconverters.

- Works-RSCM should be used for single channel up or downconverters
- Works-RSCM-Dual should be used for dual channel up or downconverters
- Works-RSCM-Triple should be used for triple channel up or downconverters
- Works-RSCM-Quad should be used for quad channel up or downconverters

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch parameters
- --- The channel parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The device address (decimal) as set at the device's front panel.
- converterType --- The units converter type. Informational only.
- extRef.x --- The external reference source of the individual unit.
  - INTERNAL
  - EXTERNAL
  - AUTO
- bandSelect.x --- The band selection of the individual unit.
  - BAND-1
  - BAND-2

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision1	TEXT	R/O	StringRange R/O
info.revision2	TEXT	R/O	StringRange R/O
config.converterType	CHOICE	SAVE SETUP	EnumRange (UNKNOWN UPCONVERTER DOWNCONVERTER)
config.extRef.0	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.1	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.2	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.3	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.4	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.5	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.6	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.7	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.8	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.bandSelect.0	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.1	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.2	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.3	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.4	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.5	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)

name	*type*	*flags*	*range*
config.bandSelect.6	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.7	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.8	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
internal.0	TEXT	R/O	StringRange R/O
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O
internal.4	TEXT	R/O	StringRange R/O
internal.5	TEXT	R/O	StringRange R/O
internal.6	TEXT	R/O	StringRange R/O
internal.7	TEXT	R/O	StringRange R/O
internal.8	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8 OFF)
location.1	CHOICE	R/O	EnumRange R/O (A B)
location.2	CHOICE	R/O	EnumRange R/O (A B)
location.3	CHOICE	R/O	EnumRange R/O (A B)
location.4	CHOICE	R/O	EnumRange R/O (A B)
location.5	CHOICE	R/O	EnumRange R/O (A B)
location.6	CHOICE	R/O	EnumRange R/O (A B)
location.7	CHOICE	R/O	EnumRange R/O (A B)
location.8	CHOICE	R/O	EnumRange R/O (A B)
chain.0.enable	CHOICE		EnumRange DIS (OFF ON)
chain.1.enable	CHOICE		EnumRange DIS (OFF ON)
chain.2.enable	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
chain.3.enable	CHOICE		EnumRange DIS (OFF ON)
chain.4.enable	CHOICE		EnumRange DIS (OFF ON)
chain.5.enable	CHOICE		EnumRange DIS (OFF ON)
chain.6.enable	CHOICE		EnumRange DIS (OFF ON)
chain.7.enable	CHOICE		EnumRange DIS (OFF ON)
chain.8.enable	CHOICE		EnumRange DIS (OFF ON)
chain.0.name	TEXT	R/O	StringRange R/O DIS
chain.1.name	TEXT	R/O	StringRange R/O DIS
chain.2.name	TEXT	R/O	StringRange R/O DIS
chain.3.name	TEXT	R/O	StringRange R/O DIS
chain.4.name	TEXT	R/O	StringRange R/O DIS
chain.5.name	TEXT	R/O	StringRange R/O DIS
chain.6.name	TEXT	R/O	StringRange R/O DIS
chain.7.name	TEXT	R/O	StringRange R/O DIS
chain.8.name	TEXT	R/O	StringRange R/O DIS
chain.0.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
ch0.on	CHOICE		EnumRange DIS (OFF ON)
ch0.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)

name	*type*	*flags*	*range*
ch0.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch0.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch0.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.on	CHOICE		EnumRange DIS (OFF ON)
ch1.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch1.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch1.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.on	CHOICE		EnumRange DIS (OFF ON)
ch2.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch2.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch2.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)

name	*type*	*flags*	*range*
ch2.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.on	CHOICE		EnumRange DIS (OFF ON)
ch3.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch3.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch3.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.on	CHOICE		EnumRange DIS (OFF ON)
ch4.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch4.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch4.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch4.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.on	CHOICE		EnumRange DIS (OFF ON)
ch5.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch5.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch5.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.on	CHOICE		EnumRange DIS (OFF ON)
ch6.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch6.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch6.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch6.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.on	CHOICE		EnumRange DIS (OFF ON)
ch7.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch7.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch7.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.on	CHOICE		EnumRange DIS (OFF ON)
ch8.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch8.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch8.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)

name	*type*	*flags*	*range*
ch8.attn.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.a	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.a	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.b	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.b	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.attn.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain.c	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.eq.c	INTEGER		IntegerRange DIS (-64 .. 64)
internal.l	CHOICE	R/O	EnumRange R/O (OFF AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fault)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fault)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Fault)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Fault)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Fault)

9.760 Work-RSCM

This page describes the device driver and the device window for the Work Microwave RSCM N:1 Redundancy Switch

- Works-RSCM should be used for single channel up or downconverters
- Works-RSCM-Dual should be used for dual channel up or downconverters
- Works-RSCM-Triple should be used for triple channel up or downconverters
- Works-RSCM-Quad should be used for quad channel up or downconverters

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The switch parameters
- --- The channel parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

- *parameter --- description*
- address --- The device address (decimal) as set at the device's front panel.
- converterType --- The units converter type. Informational only.
- extRef.x --- The external reference source of the individual unit.
  - INTERNAL
  - EXTERNAL
  - AUTO
- bandSelect.x --- The band selection of the individual unit.
  - BAND-1
  - BAND-2
- hasEqualizer --- Defines if the units are equipped with an equalizer.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.revision1	TEXT	R/O	StringRange R/O
info.revision2	TEXT	R/O	StringRange R/O
config.converterType	CHOICE	SAVE SETUP	EnumRange (UNKNOWN UPCONVERTER DOWNCONVERTER)
config.extRef.0	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.1	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.2	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.3	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.4	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.5	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.6	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.7	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.extRef.8	CHOICE	SETUP	EnumRange DIS (INTERNAL EXTERNAL AUTO)
config.bandSelect.0	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.1	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.2	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)

name	*type*	*flags*	*range*
config.bandSelect.3	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.4	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.5	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.6	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.7	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
config.bandSelect.8	CHOICE	SETUP	EnumRange DIS (BAND-1 BAND-2)
internal.0	TEXT	R/O	StringRange R/O
internal.1	TEXT	R/O	StringRange R/O
internal.2	TEXT	R/O	StringRange R/O
internal.3	TEXT	R/O	StringRange R/O
internal.4	TEXT	R/O	StringRange R/O
internal.5	TEXT	R/O	StringRange R/O
internal.6	TEXT	R/O	StringRange R/O
internal.7	TEXT	R/O	StringRange R/O
internal.8	TEXT	R/O	StringRange R/O
location	CHOICE		EnumRange (AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8 OFF)
location.1	CHOICE	R/O	EnumRange R/O (A B)
location.2	CHOICE	R/O	EnumRange R/O (A B)
location.3	CHOICE	R/O	EnumRange R/O (A B)
location.4	CHOICE	R/O	EnumRange R/O (A B)
location.5	CHOICE	R/O	EnumRange R/O (A B)
location.6	CHOICE	R/O	EnumRange R/O (A B)
location.7	CHOICE	R/O	EnumRange R/O (A B)
location.8	CHOICE	R/O	EnumRange R/O (A B)
chain.0.enable	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
chain.1.enable	CHOICE		EnumRange DIS (OFF ON)
chain.2.enable	CHOICE		EnumRange DIS (OFF ON)
chain.3.enable	CHOICE		EnumRange DIS (OFF ON)
chain.4.enable	CHOICE		EnumRange DIS (OFF ON)
chain.5.enable	CHOICE		EnumRange DIS (OFF ON)
chain.6.enable	CHOICE		EnumRange DIS (OFF ON)
chain.7.enable	CHOICE		EnumRange DIS (OFF ON)
chain.8.enable	CHOICE		EnumRange DIS (OFF ON)
chain.0.name	TEXT	R/O	StringRange R/O DIS
chain.1.name	TEXT	R/O	StringRange R/O DIS
chain.2.name	TEXT	R/O	StringRange R/O DIS
chain.3.name	TEXT	R/O	StringRange R/O DIS
chain.4.name	TEXT	R/O	StringRange R/O DIS
chain.5.name	TEXT	R/O	StringRange R/O DIS
chain.6.name	TEXT	R/O	StringRange R/O DIS
chain.7.name	TEXT	R/O	StringRange R/O DIS
chain.8.name	TEXT	R/O	StringRange R/O DIS
chain.0.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.1.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.2.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.3.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.4.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.5.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.6.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.7.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)
chain.8.chainFault	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
ch0.on	CHOICE		EnumRange DIS (OFF ON)
ch0.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch0.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch0.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.on	CHOICE		EnumRange DIS (OFF ON)
ch1.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch1.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch1.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch1.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch1.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.on	CHOICE		EnumRange DIS (OFF ON)
ch2.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch2.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch2.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch2.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch2.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.on	CHOICE		EnumRange DIS (OFF ON)
ch3.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch3.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch3.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch3.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch3.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.on	CHOICE		EnumRange DIS (OFF ON)

name	*type*	*flags*	*range*
ch4.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch4.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch4.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch4.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch4.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.on	CHOICE		EnumRange DIS (OFF ON)
ch5.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch5.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch5.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch5.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch5.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.on	CHOICE		EnumRange DIS (OFF ON)
ch6.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch6.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch6.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch6.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch6.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.on	CHOICE		EnumRange DIS (OFF ON)
ch7.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch7.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch7.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)

name	*type*	*flags*	*range*
ch7.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch7.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.on	CHOICE		EnumRange DIS (OFF ON)
ch8.attn	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.gain	FLOAT		DoubleRange DIS (0.0 .. 30.0)
ch8.freq	FLOAT		DoubleRange DIS (0.000 .. 0.000)
ch8.attnOffset	FLOAT		DoubleRange DIS (-30.0 .. 30.0)
ch8.eq	INTEGER		IntegerRange DIS (-64 .. 64)
ch8.eqOffset	INTEGER		IntegerRange DIS (-64 .. 64)
internal.l	CHOICE	R/O	EnumRange R/O (OFF AUTO CHANNEL-1 CHANNEL-2 CHANNEL-3 CHANNEL-4 CHANNEL-5 CHANNEL-6 CHANNEL-7 CHANNEL-8)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PSU1)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSU2)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Switched)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Unit 0 Fault)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Unit 1 Fault)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Unit 2 Fault)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Unit 3 Fault)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Unit 4 Fault)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Unit 5 Fault)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Unit 6 Fault)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Unit 7 Fault)
faults.18	ALARM	R/O	AlarmFlagRange R/O (Unit 8 Fault)

9.761 Work-SBD

This page describes the device driver and the device window for the Work Microwave Synthesized block downconverters SSBD.

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters
- --- The meter readings.
- The redundancy switching parameters.
- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
loFreq	Defines the LO Frequency.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.762 Work-SBU

This page describes the device driver and the device window for the Work Microwave Synthesized block upconverters SBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
loFreq	Defines the LO Frequency.
accessMode	**Full Control:** Enables changes on all tx parameters. **Blocked by RSCM:** Disables changes on all tx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)

name	*type*	*flags*	*range*
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.763 Work-SCD

This page describes the device driver and the device window for the Work Microwave SCD/HCD downconverter series with one channel.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **L4:** 1600 - 1800MHz, **S:** 2000 - 2300MHz, **C:** 3400 - 4200MHz, **C1:** 3400 - 4800MHz, **X:** 7250 - 7750MHz, **KU:** 10700 - 12750MHz, **KU4:** 9300 - 9500MHz, **KA:** 18100 - 21200MHz, **KA1:** 31800 - 32300MHz, **KA2:** 17700 - 19500MHz, **KA4:** 27500 - 31000MHz, **KA15-720:** 24000 - 27000MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L4 S C C1 X KU KU4 KA KA1 KA2 KA4 KA15-720)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.eq	INTEGER		IntegerRange DIS (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PLO)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW PLL)
faults.07	ALARM	R/O	AlarmFlagRange R/O (VHF Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (FSY Synthesizer)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter comm)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.764 Work-SCU

This page describes the device driver and the device window for the Work Microwave SCU/HCU upconverter series with one channel.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	*description*

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range: **S:** 2000 - 2300MHz, **C:** 5850 - 6650MHz, **C1:** 5850 - 7200MHz, **X:** 7900 - 8400MHz, **X3:** 7000 - 9000MHz, **KU:** 12750 - 14500MHz, **KU2:** 11800 - 13400MHz, **K:** 17300 - 18400MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (S C C1 X X3 KU KU2 K)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (12750.0000 .. 14500.0000)
tx.eq	INTEGER		IntegerRange DIS (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)

name	*type*	*flags*	*range*
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (PLO)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW PLL)
faults.07	ALARM	R/O	AlarmFlagRange R/O (VHF Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (FSY Synthesizer)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter comm)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.765 Work-SDM2-MT6-Modulator

This page describes the device driver and the device window for the Work Satcom DVB-S/S2 modulator SDM2 (L-Band version)

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	*description*
address	Device device's multipoint communication address. This is a single upper case character ('A', 'B', ...) which must match the setting made at the device's front panel.

parameter	*description*
autoSave	Setting this to ON, all settings will persist over power cycles.
muteInput	Setting this to ON enables the external mute input of the device.
alarmRelay	Setting this to ON enables the alarm relay output of the device.
Serial	The serial interface parameter of the device may be set at the front panel. The interface settings used in the sat-nms software must match the serial interface parameter of the device. A common setting is to use *19200,N,8,1* .

Network Communication

This device driver may be used to control the Work Satcom modulator/upconverter through a serial (RS232, RS422) interface or via a network connection. To enable the latter mode, set the network interface parameters of the device at it's front panel to values which are suitable to your network layout. At the device setup window , set the device's IP address / IP port instead of a tty interface name.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision.1	TEXT	R/O	StringRange R/O
info.revision.2	TEXT	R/O	StringRange R/O
info.revision.3	TEXT	R/O	StringRange R/O
info.t.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dvbmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.converter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
info.t.modchip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.fifolevel	FLOAT	R/O	DoubleRange R/O (0 .. 0)
info.t.rmsintmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dcclock	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dclo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps33	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps65	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps25	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcconv	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.muteInput	CHOICE	SETUP	EnumRange (OFF ON)
config.alarmRelay	CHOICE	SETUP	EnumRange (OFF ON)
config.banner	TEXT	SETUP	StringRange
config.extLoFreq	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.power	FLOAT		DoubleRange (-25.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mod.slopeComp	INTEGER		IntegerRange (-9 .. 9)
refClkSrc	CHOICE		EnumRange (INT EXT AUTO)
tx.mod.cwMode	CHOICE		EnumRange (OFF CW)
tx.ifc.tsMode	CHOICE		EnumRange (MULTI-TS SINGLE-TS)
tx.ifc.dualChannel	CHOICE		EnumRange (1+2 3+4 5+6)
tx.ifc.spiSync	CHOICE		EnumRange (SYNC BYTE PSYNC SIG)
tx.ifc.asiInput	CHOICE		EnumRange (1 2)
tx.ifc.asiAuto	CHOICE		EnumRange (ON OFF)
tx.ifc.actualAsi	CHOICE		EnumRange (1 2 3 4 5 6)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT)
tx.ifc.framingMode	CHOICE		EnumRange (188 204)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.rollOff	CHOICE		EnumRange (20% 25% 35%)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mod.bbFrame	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.pilots	CHOICE		EnumRange (OFF ON)
tx.mod.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.shortBchCode	CHOICE		EnumRange (ORIGINAL INDUSTRY)
tx.plScrambling	INTEGER		IntegerRange (0 .. 262141)
tx.pmtPid	INTEGER		IntegerRange (16 .. 8190)
tx.videoPid	INTEGER		IntegerRange (16 .. 8190)
tx.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.1.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.2.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.3.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.4.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.5.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.6.dataInput	CHOICE		EnumRange (DISABLED ASI)
tx.mts.1.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.2.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.3.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.4.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.5.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.6.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.1.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.2.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.3.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.4.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.5.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.6.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.1.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.2.nullPckDel	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mts.3.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.4.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.5.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.6.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.1.padding	CHOICE		EnumRange (OFF ON)
tx.mts.2.padding	CHOICE		EnumRange (OFF ON)
tx.mts.3.padding	CHOICE		EnumRange (OFF ON)
tx.mts.4.padding	CHOICE		EnumRange (OFF ON)
tx.mts.5.padding	CHOICE		EnumRange (OFF ON)
tx.mts.6.padding	CHOICE		EnumRange (OFF ON)
tx.mts.1.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.2.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.3.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.4.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.5.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.6.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.1.genericAuto	TEXT		StringRange
tx.mts.2.genericAuto	TEXT		StringRange
tx.mts.3.genericAuto	TEXT		StringRange
tx.mts.4.genericAuto	TEXT		StringRange
tx.mts.5.genericAuto	TEXT		StringRange
tx.mts.6.genericAuto	TEXT		StringRange

name	*type*	*flags*	*range*
tx.mts.1.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mts.2.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mts.3.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)

name	*type*	*flags*	*range*
tx.mts.4.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mts.5.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mts.6.type	CHOICE		EnumRange (S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mts.1.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.2.pilots	CHOICE		EnumRange (OFF ON GLOBAL)

name	*type*	*flags*	*range*
tx.mts.3.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.4.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.5.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.6.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.1.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.2.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.3.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.4.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.5.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.6.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.1.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.2.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.3.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.4.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.5.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.6.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.1.upl	TEXT		StringRange
tx.mts.2.upl	TEXT		StringRange
tx.mts.3.upl	TEXT		StringRange
tx.mts.4.upl	TEXT		StringRange
tx.mts.5.upl	TEXT		StringRange
tx.mts.6.upl	TEXT		StringRange
tx.mts.1.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.2.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.3.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.4.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.5.mode	CHOICE		EnumRange (VCM/ACM CCM)

name	*type*	*flags*	*range*
tx.mts.6.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.1.tsgs	TEXT		StringRange
tx.mts.2.tsgs	TEXT		StringRange
tx.mts.3.tsgs	TEXT		StringRange
tx.mts.4.tsgs	TEXT		StringRange
tx.mts.5.tsgs	TEXT		StringRange
tx.mts.6.tsgs	TEXT		StringRange
tx.mts.1.generic	TEXT		StringRange
tx.mts.2.generic	TEXT		StringRange
tx.mts.3.generic	TEXT		StringRange
tx.mts.4.generic	TEXT		StringRange
tx.mts.5.generic	TEXT		StringRange
tx.mts.6.generic	TEXT		StringRange
tx.mts.1.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.2.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.3.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.4.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.5.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.6.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.1.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.2.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.3.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.4.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)

name	*type*	*flags*	*range*
tx.mts.5.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.6.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.1.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.2.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.3.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.4.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.5.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.6.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.1.mergerFullThr	TEXT		StringRange
tx.mts.2.mergerFullThr	TEXT		StringRange
tx.mts.3.mergerFullThr	TEXT		StringRange
tx.mts.4.mergerFullThr	TEXT		StringRange
tx.mts.5.mergerFullThr	TEXT		StringRange
tx.mts.6.mergerFullThr	TEXT		StringRange
tx.mts.1.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.2.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.3.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.4.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.5.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.6.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.1.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.2.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.3.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.4.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.5.limiter	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mts.6.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.1.framingMode	CHOICE		EnumRange (188 204)
tx.mts.2.framingMode	CHOICE		EnumRange (188 204)
tx.mts.3.framingMode	CHOICE		EnumRange (188 204)
tx.mts.4.framingMode	CHOICE		EnumRange (188 204)
tx.mts.5.framingMode	CHOICE		EnumRange (188 204)
tx.mts.6.framingMode	CHOICE		EnumRange (188 204)
tx.mts.1.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.2.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.3.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.4.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.5.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.6.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.1.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.2.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.3.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.4.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.5.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.6.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.1.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.2.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.3.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.4.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.5.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.6.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.1.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.2.tsoip	CHOICE		EnumRange (ASI TSOIP)

name	*type*	*flags*	*range*
tx.mts.3.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.4.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.5.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.6.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.1.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.2.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.3.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.4.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.5.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.6.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.1.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.2.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.3.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.4.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.5.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.6.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.1.red.location	CHOICE		EnumRange (1 2)
tx.mts.2.red.location	CHOICE		EnumRange (1 2)
tx.mts.3.red.location	CHOICE		EnumRange (3 4)
tx.mts.4.red.location	CHOICE		EnumRange (3 4)
tx.mts.5.red.location	CHOICE		EnumRange (5 6)
tx.mts.6.red.location	CHOICE		EnumRange (5 6)
tx.mts.1.actualAsi	CHOICE	R/O	EnumRange R/O (1 2)
tx.mts.2.actualAsi	CHOICE	R/O	EnumRange R/O (1 2)
tx.mts.3.actualAsi	CHOICE	R/O	EnumRange R/O (3 4)

name	*type*	*flags*	*range*
tx.mts.4.actualAsi	CHOICE	R/O	EnumRange R/O (3 4)
tx.mts.5.actualAsi	CHOICE	R/O	EnumRange R/O (5 6)
tx.mts.6.actualAsi	CHOICE	R/O	EnumRange R/O (5 6)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.si	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mz	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.ac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mhx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.tx.on	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RAM error)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interpolator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Clock PLL overload)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Security device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (FIFO full)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Clock oscillator)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Local oscillator)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PLL lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter)

name	*type*	*flags*	*range*
faults.10	ALARM	R/O	AlarmFlagRange R/O (UPC reference)
faults.11	ALARM	R/O	AlarmFlagRange R/O (UPC communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DVB communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TS clock osc.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Transmit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.16	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (DVB input sync)
faults.18	ALARM	R/O	AlarmFlagRange R/O (ASI B signal low)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PCR memory)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.21	ALARM	R/O	AlarmFlagRange R/O (ASI In-A missing)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ASI In-B missing)
faults.30	ALARM	R/O	AlarmFlagRange R/O (MTS1 input sync loss)
faults.31	ALARM	R/O	AlarmFlagRange R/O (MTS1 config invalid)
faults.32	ALARM	R/O	AlarmFlagRange R/O (MTS1 config auth)
faults.33	ALARM	R/O	AlarmFlagRange R/O (MTS1 sync buffer full)
faults.35	ALARM	R/O	AlarmFlagRange R/O (MTS1 biss buffer full)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (MTS1 channel shutdown)
faults.37	ALARM	R/O	AlarmFlagRange R/O (MTS1 channel disabled)
faults.38	ALARM	R/O	AlarmFlagRange R/O (MTS1 merger buffer full)
faults.39	ALARM	R/O	AlarmFlagRange R/O (MTS1 signal lost)
faults.40	ALARM	R/O	AlarmFlagRange R/O (MTS2 input sync loss)
faults.41	ALARM	R/O	AlarmFlagRange R/O (MTS2 config invalid)
faults.42	ALARM	R/O	AlarmFlagRange R/O (MTS2 config auth)
faults.43	ALARM	R/O	AlarmFlagRange R/O (MTS2 sync buffer full)
faults.45	ALARM	R/O	AlarmFlagRange R/O (MTS2 biss buffer full)
faults.46	ALARM	R/O	AlarmFlagRange R/O (MTS2 channel shutdown)
faults.47	ALARM	R/O	AlarmFlagRange R/O (MTS2 channel disabled)
faults.48	ALARM	R/O	AlarmFlagRange R/O (MTS2 merger buffer full)
faults.49	ALARM	R/O	AlarmFlagRange R/O (MTS2 signal lost)
faults.50	ALARM	R/O	AlarmFlagRange R/O (MTS3 input sync loss)
faults.51	ALARM	R/O	AlarmFlagRange R/O (MTS3 config invalid)
faults.52	ALARM	R/O	AlarmFlagRange R/O (MTS3 config auth)

name	*type*	*flags*	*range*
faults.53	ALARM	R/O	AlarmFlagRange R/O (MTS3 sync buffer full)
faults.55	ALARM	R/O	AlarmFlagRange R/O (MTS3 biss buffer full)
faults.56	ALARM	R/O	AlarmFlagRange R/O (MTS3 channel shutdown)
faults.57	ALARM	R/O	AlarmFlagRange R/O (MTS3 channel disabled)
faults.58	ALARM	R/O	AlarmFlagRange R/O (MTS3 merger buffer full)
faults.59	ALARM	R/O	AlarmFlagRange R/O (MTS3 signal lost)
faults.60	ALARM	R/O	AlarmFlagRange R/O (MTS4 input sync loss)
faults.61	ALARM	R/O	AlarmFlagRange R/O (MTS4 config invalid)
faults.62	ALARM	R/O	AlarmFlagRange R/O (MTS4 config auth)
faults.63	ALARM	R/O	AlarmFlagRange R/O (MTS4 sync buffer full)
faults.65	ALARM	R/O	AlarmFlagRange R/O (MTS4 biss buffer full)
faults.66	ALARM	R/O	AlarmFlagRange R/O (MTS4 channel shutdown)
faults.67	ALARM	R/O	AlarmFlagRange R/O (MTS4 channel disabled)
faults.68	ALARM	R/O	AlarmFlagRange R/O (MTS4 merger buffer full)
faults.69	ALARM	R/O	AlarmFlagRange R/O (MTS4 signal lost)
faults.70	ALARM	R/O	AlarmFlagRange R/O (MTS5 input sync loss)

name	*type*	*flags*	*range*
faults.71	ALARM	R/O	AlarmFlagRange R/O (MTS5 config invalid)
faults.72	ALARM	R/O	AlarmFlagRange R/O (MTS5 config auth)
faults.73	ALARM	R/O	AlarmFlagRange R/O (MTS5 sync buffer full)
faults.75	ALARM	R/O	AlarmFlagRange R/O (MTS5 biss buffer full)
faults.76	ALARM	R/O	AlarmFlagRange R/O (MTS5 channel shutdown)
faults.77	ALARM	R/O	AlarmFlagRange R/O (MTS5 channel disabled)
faults.78	ALARM	R/O	AlarmFlagRange R/O (MTS5 merger buffer full)
faults.79	ALARM	R/O	AlarmFlagRange R/O (MTS5 signal lost)
faults.80	ALARM	R/O	AlarmFlagRange R/O (MTS6 input sync loss)
faults.81	ALARM	R/O	AlarmFlagRange R/O (MTS6 config invalid)
faults.82	ALARM	R/O	AlarmFlagRange R/O (MTS6 config auth)
faults.83	ALARM	R/O	AlarmFlagRange R/O (MTS6 sync buffer full)
faults.85	ALARM	R/O	AlarmFlagRange R/O (MTS6 biss buffer full)
faults.86	ALARM	R/O	AlarmFlagRange R/O (MTS6 channel shutdown)
faults.87	ALARM	R/O	AlarmFlagRange R/O (MTS6 channel disabled)
faults.88	ALARM	R/O	AlarmFlagRange R/O (MTS6 merger buffer full)

name	*type*	*flags*	*range*
faults.89	ALARM	R/O	AlarmFlagRange R/O (MTS6 signal lost)

9.766 Work-SDM2-Modulator

This page describes the device driver and the device window for the Work Satcom DVB-S/S2 modulator SDM2 (L-Band version) with Option MT6 (6 ASI inputs).

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	*description*
address	Device device's multipoint communication address. This is a single upper case character ('A', 'B', ...) which must match the setting made at the device's front panel.
autoSave	Setting this to ON, all settings will persist over power cycles.
muteInput	Setting this to ON enables the external mute input of the device.
alarmRelay	Setting this to ON enables the alarm relay output of the device.
Serial	The serial interface parameter of the device may be set at the front panel. The interface settings used in the sat-nms software must match the serial interface parameter of the device. A common setting is to use *19200,N,8,1* .

Network Communication

This device driver may be used to control the Work Satcom modulator/upconverter through a serial (RS232, RS422) interface or via a network connection. To enable the latter mode, set the

network interface parameters of the device at it's front panel to values which are suitable to your network layout. At the device setup window , set the device's IP address / IP port instead of a tty interface name.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.band	TEXT	R/O	StringRange R/O
info.revision.1	TEXT	R/O	StringRange R/O
info.revision.2	TEXT	R/O	StringRange R/O
info.revision.3	TEXT	R/O	StringRange R/O
info.t.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dvbmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.converter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.modchip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.fifolevel	FLOAT	R/O	DoubleRange R/O (0 .. 0)
info.t.rmsintmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dcclock	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dclo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps33	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps65	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
info.t.dcps21	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps12	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcps25	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.t.dcconv	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.muteInput	CHOICE	SETUP	EnumRange (OFF ON)
config.alarmRelay	CHOICE	SETUP	EnumRange (OFF ON)
config.banner	TEXT	SETUP	StringRange
config.extLoFreq	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.power	FLOAT		DoubleRange (-25.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mod.slopeComp	INTEGER		IntegerRange (-9 .. 9)
refClkSrc	CHOICE		EnumRange (INT EXT AUTO)
tx.mod.cwMode	CHOICE		EnumRange (CW MOD)
tx.ifc.spiSync	CHOICE		EnumRange (SYNC BYTE PSYNC SIG)
tx.ifc.asiInput	CHOICE		EnumRange (A B)
tx.ifc.asiAuto	CHOICE		EnumRange (ON OFF)
tx.ifc.actualAsi	CHOICE		EnumRange (A B)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT)
tx.ifc.framingMode	CHOICE		EnumRange (188 204)
tx.ifc.dataInput	CHOICE		EnumRange (ASI SPI)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.rollOff	CHOICE		EnumRange (20% 25% 35%)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)
tx.mod.bbFrame	CHOICE		EnumRange (OFF ON)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.shortBchCode	CHOICE		EnumRange (ORIGINAL INDUSTRY)
tx.plScrambling	INTEGER		IntegerRange (0 .. 262141)
tx.pmtPid	INTEGER		IntegerRange (16 .. 8190)
tx.videoPid	INTEGER		IntegerRange (16 .. 8190)
tx.stillPicture	CHOICE		EnumRange (OFF ON)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.si	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mz	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.ac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.on	TEXT	R/O	StringRange R/O
internal.min	FLOAT		DoubleRange (0.00000 .. 0.00000)
internal.max	FLOAT		DoubleRange (0.00000 .. 0.00000)
faults.01	ALARM	R/O	AlarmFlagRange R/O (RAM error)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interpolator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Clock PLL overload)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Security device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (FIFO full)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Clock oscillator)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Local oscillator)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PLL lock)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.10	ALARM	R/O	AlarmFlagRange R/O (UPC reference)
faults.11	ALARM	R/O	AlarmFlagRange R/O (UPC communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DVB communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TS clock osc.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Transmit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.16	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (DVB input sync)
faults.18	ALARM	R/O	AlarmFlagRange R/O (ASI B signal low)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PCR memory)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.21	ALARM	R/O	AlarmFlagRange R/O (ASI In-A missing)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ASI In-B missing)

9.767 Work-SM2CU-Modulator

This page describes the device driver and the device window for the Work Satcom modulator / upconverter devices SM2CU / HM2CU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device settings.

- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	***description***
address	Device device's multipoint communication address. This is a single upper case character ('A', 'B', ...) which must match the setting made at the device's front panel.
autoSave	Setting this to ON, all settings will persist over power cycles.
muteInput	Setting this to ON enables the external mute input of the device.
alarmRelay	Setting this to ON enables the alarm relay output of the device.
Serial	The serial interface parameter of the device may be set at the front panel. The interface settings used in the sat-nms software must match the serial interface parameter of the device. A common setting is to use *19200,N,8,1* .

Network Communication

This device driver may be used to control the Work Satcom modulator/upconverter through a serial (RS232, RS422) interface or via a network connection. To enable the latter mode, set the network interface parameters of the device at it's front panel to values which are suitable to your network layout. At the device setup window , set the device's IP address / IP port instead of a tty interface name.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision.1	TEXT	R/O	StringRange R/O
info.revision.2	TEXT	R/O	StringRange R/O
info.revision.3	TEXT	R/O	StringRange R/O
info.t.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dvbmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.converter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.modchip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.muteInput	CHOICE	SETUP	EnumRange (OFF ON)
config.alarmRelay	CHOICE	SETUP	EnumRange (OFF ON)
config.banner	TEXT	SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.streamId	TEXT	SAVE SETUP	StringRange
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.power	FLOAT		DoubleRange (-30.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
tx.mod.symbolRate	FLOAT	NOPRESET	DoubleRange (0.000000 .. 0.000000)

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10)
tx.mod.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mod.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 16QAM 16APSK 32APSK)
tx.mod.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5)
tx.mod.slopeComp	INTEGER		IntegerRange (-9 .. 9)
refClkSrc	CHOICE		EnumRange (INT EXT AUTO)
tx.mod.cwMode	CHOICE		EnumRange (OFF CW)
tx.ifc.spiSync	CHOICE		EnumRange (SYNC BYTE PSYNC SIG)
tx.ifc.asiInput	CHOICE		EnumRange (A B)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT)
tx.ifc.framingMode	CHOICE		EnumRange (188 204)
tx.ifc.dataInput	CHOICE		EnumRange (ASI SPI)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.mod.rollOff	CHOICE		EnumRange (20% 25% 35%)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.bbFrame	CHOICE		EnumRange (OFF ON)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)
tx.mod.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.shortBchCode	CHOICE		EnumRange (ORIGINAL INDUSTRY)
tx.plScrambling	INTEGER		IntegerRange (0 .. 262141)
tx.pmtPid	INTEGER		IntegerRange (16 .. 8190)
tx.videoPid	INTEGER		IntegerRange (16 .. 8190)
tx.stillPicture	CHOICE		EnumRange (OFF ON)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mz	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tx.on	TEXT	R/O	StringRange R/O
internal.type	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (RAM error)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interpolator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Clock PLL overload)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Security device)
faults.05	ALARM	R/O	AlarmFlagRange R/O (FIFO full)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Clock oscillator)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Local oscillator)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PLL lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.10	ALARM	R/O	AlarmFlagRange R/O (UPC reference)

name	*type*	*flags*	*range*
faults.11	ALARM	R/O	AlarmFlagRange R/O (UPC communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DVB communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TS clock osc.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Transmit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.16	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (DVB input sync)
faults.18	ALARM	R/O	AlarmFlagRange R/O (ASI B signal)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PCR memory)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Configuration)

9.768 Work-SSBD

This page describes the device driver and the device window for the Work Microwave Synthesized block downconverters SSBD.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)

name	*type*	*flags*	*range*
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.769 Work-SSBU

This page describes the device driver and the device window for the Work Microwave Synthesized block upconverters SSBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The operational parameters
- --- The redundancy switching parameters

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.770 Work-STLT2-KuK

This page describes the device driver and the device window for the Work Microwave Test Loop Translator with LO frequency setting.

The driver was tested with the following models:

- STLT2-KuK

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
accessMode	**Full Control:** Enables changes on all tx parameters. **Blocked by RSCM:** Disables changes on all tx parameters (readonly).

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.converter	TEXT	R/O	StringRange R/O
info.frontPanel	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL alarm)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (PLO alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer alarm)
faults.08	ALARM	R/O	AlarmFlagRange R/O (External ref. source)
faults.09	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX OFF)

9.771 Work-TLT-LO

This page describes the device driver and the device window for the Work Microwave Test Loop Translator with LO frequency setting.

The driver was tested with the following models: VSTLT-K-S001

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.

parameter	*description*
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
accessMode	**Full Control:** Enables changes on all tx parameters. **Blocked by RSCM:** Disables changes on all tx parameters (readonly).

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)

name	*type*	*flags*	*range*
config.band	CHOICE	SAVE SETUP	EnumRange (NONE S C C1 X KU KU1 K KA1 KA)
config.deviceComment	TEXT	SAVE SETUP	StringRange
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer alarm)
faults.09	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.10	ALARM	R/O	AlarmFlagRange R/O (External mute input)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature high)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature low)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX OFF)

9.772 Work-TLT

This page describes the device driver and the device window for the Work Microwave Test Loop Translator.

The driver was tested with the following models:

- VHTLT-Ku-OD-RIN-S0078
- VSTLT-X-S00306

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the unit.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
adjLO	Defines the usage of a individual transmit frequency.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.adjLO	CHOICE	SAVE SETUP	EnumRange (1 2)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.rxBand	CHOICE	SAVE SETUP	EnumRange (NONE S C C1 X KU KU1 K KA1 KA)
config.txBand	CHOICE	SAVE SETUP	EnumRange (NONE S C C1 X KU KU1 K KA1 KA)
config.deviceComment	TEXT	SAVE SETUP	StringRange
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (0.0000 .. 0.0000)
tx.frequency	FLOAT		DoubleRange DIS (0.0000 .. 0.0000)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal alarm)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL alarm)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier alarm)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output alarm)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer alarm)

name	*type*	*flags*	*range*
faults.09	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.10	ALARM	R/O	AlarmFlagRange R/O (External mute input)
faults.11	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.12	ALARM	R/O	AlarmFlagRange R/O (Temperature high)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Temperature low)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX OFF)

9.773 Work-TriSBD

This page describes the device driver and the device window for the Work Microwave 3-channel block downconverters

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters.
- --- The meter readings.
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
loFreq.3	Defines the LO Frequency for Band 3.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port = 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.3	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.3	FLOAT		DoubleRange (0.0 .. 30.0)

name	*type*	*flags*	*range*
rx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
rx.eq.3	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.774 Work-TriSBU

This page describes the device driver and the device window for the Work Microwave Synthesized triple / 3-channel block upconverters VSBU.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
loFreq.1	Defines the LO Frequency for Band 1.
loFreq.2	Defines the LO Frequency for Band 2.
loFreq.3	Defines the LO Frequency for Band 3.
accessMode	**Full Control:** Enables changes on all tx parameters**Blocked by RSCM:** Disables changes on all tx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port = 50505

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.loFreq.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.loFreq.3	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
tx.on	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
tx.gain.3	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn.3	FLOAT		DoubleRange (0.0 .. 30.0)
tx.eq.1	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.2	FLOAT		DoubleRange DIS (0.0 .. 8.0)
tx.eq.3	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.775 Work-TripleVSCD

This page describes the device driver and the device window for the Work Microwave VSCD/VHCD downconverter series with three channels.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings
- --- The redundancy parameters
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **L4:** 1600 - 1800MHz, **S:** 2000 - 2300MHz, **C:** 3400 - 4200MHz, **C1:** 3400 - 4800MHz, **X:** 7250 - 7750MHz, **KU:** 10700 - 12750MHz, **KU4:** 9300 - 9500MHz, **KA:** 18100 - 21200MHz, **KA1:** 31800 - 32300MHz, **KA2:** 17700 - 19500MHz, **KA4:** 27500 - 31000MHz, **KA15-720:** 24000 - 27000MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.
singleChanMute	Defines if only a global mute can be applied = YES, or if each channel can also be muted individually = NO.

parameter	*description*
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)

name	*type*	*flags*	*range*
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L2 L4 S S1 S4 C C1 X X3 KU KU2 KU4 KA KA1 KA2 KA3 KA4 KA7 KA21 KA15-720)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.singleChanMute	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (OFF ON)
rx.on.1	CHOICE		EnumRange (OFF ON)
rx.on.2	CHOICE		EnumRange (OFF ON)
rx.on.3	CHOICE		EnumRange (OFF ON)
rx.gain.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.1	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.2	FLOAT		DoubleRange (0.0 .. 30.0)
rx.gain.3	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn.3	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.eq.1	INTEGER		IntegerRange (-64 .. 64)
rx.eq.2	INTEGER		IntegerRange (-64 .. 64)
rx.eq.3	INTEGER		IntegerRange (-64 .. 64)

name	*type*	*flags*	*range*
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.776 Work-VCR

This page describes the device driver and the device window for the Work Remote Control Unit VRC for Up/Downconverters. It is a monitoring only driver as addition to the existing device drivers for the respective Up and Downconverter units.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the controller.

parameter	***description***
address	The device address (Default = A)

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.BaseSystemVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.sa0	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Converter Com)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (External Mute)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EEPROM)
faults.05	ALARM	R/O	AlarmFlagRange R/O (High Temp.)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Low Temp.)
faults.07	ALARM	R/O	AlarmFlagRange R/O (NTP)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Summary)

9.777 Work-VSCD

This page describes the device driver and the device window for the Work Microwave VSCD/VHCD downconverter series.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.

parameter	*description*
Serial	Remember to set the line parameters for the serial interface.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **L4:** 1600 - 1800MHz, **S:** 2000 - 2300MHz, **C:** 3400 - 4200MHz, **C1:** 3400 - 4800MHz, **X:** 7250 - 7750MHz, **KU:** 10700 - 12750MHz, **KU4:** 9300 - 9500MHz, **KA:** 18100 - 21200MHz, **KA1:** 31800 - 32300MHz, **KA2:** 17700 - 19500MHz, **KA4:** 27500 - 31000MHz, **KA15-720:** 24000 - 27000MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L2 L4 S S1 S4 C C1 X X3 KU KU2 KU4 KA KA1 KA2 KA3 KA4 KA7 KA21 KA15-720)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (10700.0000 .. 12750.0000)
rx.eq	INTEGER		IntegerRange DIS (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.778 Work-VSCSR

This page describes the device driver and the device window for the Work Microwave Calibration Source Type VSCSR

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	*description*
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
tx.on	CHOICE	NOPRESET	EnumRange (OFF ON)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.779 Work-VSCU

This page describes the device driver and the device window for the Work Microwave

VSCU/VHCU upconverter series.

Device window pages

The following table shows which device window pages are available with this individual device type.

- --- The control parameters
- --- The meter readings.
- --- The redundancy parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
band	Defines the used frequency band and sets the corresponding rx frequency range **S:** 2000 - 2300MHz, **C:** 5850 - 6650MHz, **C1:** 5850 - 7200MHz, **X:** 7900 - 8400MHz, **X3:** 7000 - 9000MHz, **KU:** 12750 - 14500MHz, **K:** 17300 - 18400MHz.
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	***type***	***flags***	***range***

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.IfModuleVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.band	CHOICE	SAVE SETUP	EnumRange (L2 S S4 C C1 X X3 X4 X6 KU KU1 KU4 K KA KA1 KA2 KA3 KA5 KA21)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (12750.0000 .. 14500.0000)
tx.eq	INTEGER		IntegerRange DIS (-64 .. 64)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
info.temp.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.controller	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.refBoard	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.temp.synthesizer	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)

name	*type*	*flags*	*range*
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)
internal.attn	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

9.780 Work-VSM2CU-Modulator

This page describes the device driver and the device window for the Work Satcom modulator / upconverter devices VSM2CU.

This driver includes Carried ID parameter. Not included IP input / output modules.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The device settings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device. Beside this, some installation parameters of the device may be controlled.

parameter	*description*
address	Device device's multipoint communication address. This is a single upper case character ('A', 'B', ...) which must match the setting made at the device's front panel.
autoSave	Setting this to ON, all settings will persist over power cycles.
muteInput	Setting this to ON enables the external mute input of the device.
alarmRelay	Setting this to ON enables the alarm relay output of the device.
banner	Front panel banner
extLoFreq	LO frequency of transmit equipment

parameter	*description*
Serial	The serial interface parameter of the device may be set at the front panel. The interface settings used in the sat-nms software must match the serial interface parameter of the device. A common setting is to use *19200,N,8,1* .

Network Communication

This device driver may be used to control the Work Satcom modulator/upconverter through a serial (RS232, RS422) interface or via a network connection. To enable the latter mode, set the network interface parameters of the device at it's front panel to values which are suitable to your network layout. At the device setup window , set the device's IP address / IP port instead of a tty interface name.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.modulatorVersion	TEXT	R/O	StringRange R/O
info.converterVersion	TEXT	R/O	StringRange R/O
info.frontpanelVersion	TEXT	R/O	StringRange R/O
info.carrierId	TEXT	R/O	StringRange R/O
info.t.frontpanel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.dvbmod	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.converter	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.t.modchip	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
info.fifolevel	FLOAT	R/O	DoubleRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.muteInput	CHOICE	SETUP	EnumRange (OFF ON)
config.alarmRelay	CHOICE	SETUP	EnumRange (OFF ON)
config.banner	TEXT	SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (0 .. 10000)
config.extLoFreq	FLOAT	SETUP	DoubleRange (0.0 .. 0.0)
config.lowLevel	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.highLevel	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.optionTI	CHOICE	SAVE SETUP	EnumRange (N/A TI-1 TI-2)
config.optionMT	CHOICE	SAVE SETUP	EnumRange (N/A MT-2 MT-6)
config.ratePriority	CHOICE	SAVE SETUP	EnumRange (SYMBOLRATE DATARATE)
config.streamId	TEXT	SAVE SETUP	StringRange
tx.frequency	FLOAT		DoubleRange (0.00000 .. 0.00000)
tx.power	FLOAT		DoubleRange (-25.0 .. 5.0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.mod.dataRate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mod.symbolRate	FLOAT		DoubleRange (0.000000 .. 0.000000)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3

name	*type*	*flags*	*range*
tx.mod.type	CHOICE		S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
tx.mod.standard	CHOICE	R/O	EnumRange R/O (DVB-S DVB-S2 DVB-S2X)

name	*type*	*flags*	*range*
tx.mod.modulation	CHOICE	R/O	EnumRange R/O (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mod.fec	CHOICE	R/O	EnumRange R/O (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mod.slopeComp	INTEGER		IntegerRange (-9 .. 9)
refClkSrc	CHOICE		EnumRange (INT EXT AUTO)
tx.mod.cwMode	CHOICE		EnumRange (OFF CW)
tx.ifc.spiSync	CHOICE		EnumRange (SYNC BYTE PSYNC SIG)
tx.ifc.actualAsi	CHOICE	R/O	EnumRange R/O (A B --)
flags.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.input.a	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.input.b	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.mod.clockSource	CHOICE		EnumRange (INT EXT)
tx.ifc.framingMode	CHOICE		EnumRange (188 204)
tx.ifc.dataInput	CHOICE		EnumRange (ASI-A ASI-B ASI-AUTO SPI)
protection	CHOICE	R/O	EnumRange R/O (OFF ON)
tx.mod.spectrumInvert	CHOICE		EnumRange (NORM INV)
tx.ifc.switchMode	CHOICE		EnumRange (EL-Fault TS-Auto UP-Auto TS-Manual UP-Manual)
tx.ifc.switchDelay	FLOAT		DoubleRange (0 .. 6553)
tx.mod.rollOff	CHOICE		EnumRange (20% 25% 35% 5% 10% 15%)
tx.mod.rollOffSignal	CHOICE		EnumRange (ON OFF)
tx.mod.nullPacketDelete	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mod.stuffing	CHOICE		EnumRange (all zero packet null packet)
tx.mod.bbFrame	CHOICE		EnumRange (OFF ON)
tx.mod.pilots	CHOICE		EnumRange (OFF ON)
tx.mod.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mod.shortBchCode	CHOICE		EnumRange (ORIGINAL INDUSTRY)
tx.ifc.asiEqualization	CHOICE		EnumRange DIS (OFF ON)
tx.ifc.asiPolarity	CHOICE		EnumRange DIS (OFF ON)
tx.ifc.asiCompensation	CHOICE		EnumRange DIS (OFF ON)
tx.plScrambling	INTEGER		IntegerRange (0 .. 262141)
tx.pmtPid	INTEGER		IntegerRange (16 .. 8190)
tx.videoPid	INTEGER		IntegerRange (16 .. 8190)
tx.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mux.cid.enable	CHOICE	NOPRESET	EnumRange (OFF ON)
tx.mux.cid.carrierId	TEXT	R/O	StringRange R/O
tx.mux.cid.telnNo	TEXT	NOPRESET	StringRange
tx.mux.cid.longitude	FLOAT	NOPRESET	DoubleRange (-180.000000 .. 180.000000)
tx.mux.cid.userInfo	TEXT	NOPRESET	StringRange
tx.mux.cid.latitude	FLOAT	NOPRESET	DoubleRange (-90.000000 .. 90.000000)
ipin.1.enable	CHOICE		EnumRange (ON OFF)
ipin.1.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.1.address	TEXT		StringRange
ipin.1.fecEnable	CHOICE		EnumRange (ON OFF)
ipin.1.port	INTEGER		IntegerRange (0 .. 65535)
ipin.1.delay	INTEGER		IntegerRange (1 .. 9942)

name	*type*	*flags*	*range*
ipin.1.rateEstimation	CHOICE		EnumRange (USE PCR OFF+BUFFER WITHOUT PCR OFF)
ipin.1.packetsize	CHOICE		EnumRange (188 204 AUTO)
ipin.1.fecDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.state	TEXT	R/O	StringRange R/O
ipin.1.protocol	TEXT	R/O	StringRange R/O
ipin.1.rx.packetsize	TEXT	R/O	StringRange R/O
ipin.1.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.1.pcr	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.1.rateChange	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.jitterError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.lockError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.delayFactor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.1.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.enable	CHOICE		EnumRange (ON OFF)
ipin.2.mode	CHOICE		EnumRange (MULTICAST UNICAST)
ipin.2.address	TEXT		StringRange
ipin.2.fecEnable	CHOICE		EnumRange (ON OFF)
ipin.2.port	INTEGER		IntegerRange (0 .. 65535)
ipin.2.delay	INTEGER		IntegerRange (1 .. 9942)

name	*type*	*flags*	*range*
ipin.2.rateEstimation	CHOICE		EnumRange (USE PCR OFF+BUFFER WITHOUT PCR OFF)
ipin.2.packetsize	CHOICE		EnumRange (188 204 AUTO)
ipin.2.fecDelay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecRows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.fecColumns	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.jitter	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsLost	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsUnCorr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.packetsFrame	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.state	TEXT	R/O	StringRange R/O
ipin.2.protocol	TEXT	R/O	StringRange R/O
ipin.2.rx.packetsize	TEXT	R/O	StringRange R/O
ipin.2.rx.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
ipin.2.pcr	BOOLEAN	R/O	BooleanRange R/O (true false)
ipin.2.rateChange	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.jitterError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.lockError	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.delayFactor	INTEGER	R/O	IntegerRange R/O (0 .. 0)
ipin.2.tx.datarate	FLOAT	R/O	DoubleRange R/O (0.000000 .. 0.000000)
tx.mts.channels	CHOICE		EnumRange (1/2 3/4 5/6)
tx.mts.channelMode	CHOICE		EnumRange (SINGLE MULTI)
tx.mts.1.dataInput	CHOICE		EnumRange (DISABLED ENABLED)
tx.mts.2.dataInput	CHOICE		EnumRange (DISABLED ENABLED)
tx.mts.3.dataInput	CHOICE		EnumRange (DISABLED ENABLED)

name	*type*	*flags*	*range*
tx.mts.4.dataInput	CHOICE		EnumRange (DISABLED ENABLED)
tx.mts.5.dataInput	CHOICE		EnumRange (DISABLED ENABLED)
tx.mts.6.dataInput	CHOICE		EnumRange (DISABLED ENABLED)
tx.mts.1.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.2.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.3.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.4.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.5.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.6.issyInsert	CHOICE		EnumRange (OFF ON)
tx.mts.1.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.2.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.3.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.4.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.5.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.6.issyLength	CHOICE		EnumRange (2-BYTE 3-BYTE)
tx.mts.1.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.2.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.3.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.4.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.5.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.6.nullPckDel	CHOICE		EnumRange (OFF ON)
tx.mts.1.padding	CHOICE		EnumRange (OFF ON)
tx.mts.2.padding	CHOICE		EnumRange (OFF ON)
tx.mts.3.padding	CHOICE		EnumRange (OFF ON)
tx.mts.4.padding	CHOICE		EnumRange (OFF ON)
tx.mts.5.padding	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mts.6.padding	CHOICE		EnumRange (OFF ON)
tx.mts.1.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.2.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.3.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.4.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.5.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.6.isi	INTEGER		IntegerRange (0 .. 255)
tx.mts.1.genericAuto	TEXT		StringRange
tx.mts.2.genericAuto	TEXT		StringRange
tx.mts.3.genericAuto	TEXT		StringRange
tx.mts.4.genericAuto	TEXT		StringRange
tx.mts.5.genericAuto	TEXT		StringRange
tx.mts.6.genericAuto	TEXT		StringRange
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L

name	type	flags	range
tx.mts.1.type	CHOICE		S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-

name	*type*	*flags*	*range*
			8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36
tx.mts.2.type	CHOICE		S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-

name	*type*	*flags*	*range*
tx.mts.3.type	CHOICE		32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4

name	*type*	*flags*	*range*
			S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-
tx.mts.4.type	CHOICE		32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-

name	*type*	*flags*	*range*
			8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-
tx.mts.5.type	CHOICE		9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
			EnumRange (S-BPSK-1/2 S-BPSK-2/3 S-BPSK-3/4 S-BPSK-5/6 S-BPSK-6/7 S-BPSK-7/8 S-QPSK-1/2 S-QPSK-2/3 S-QPSK-3/4 S-QPSK-5/6 S-QPSK-6/7 S-QPSK-7/8 S-8PSK-2/3 S-8PSK-5/6 S-8PSK-8/9 S-16QAM-3/4 S-16QAM-7/8 S2-QPSK-1/4 S2-QPSK-1/3 S2-QPSK-2/5 S2-QPSK-

name	*type*	*flags*	*range*
tx.mts.6.type	CHOICE		1/2 S2-QPSK-3/5 S2-QPSK-2/3 S2-QPSK-3/4 S2-QPSK-4/5 S2-QPSK-5/6 S2-QPSK-8/9 S2-QPSK-9/10 S2-8PSK-3/5 S2-8PSK-2/3 S2-8PSK-3/4 S2-8PSK-5/6 S2-8PSK-8/9 S2-8PSK-9/10 S2-16APSK-2/3 S2-16APSK-3/4 S2-16APSK-4/5 S2-16APSK-5/6 S2-16APSK-8/9 S2-16APSK-9/10 S2-32APSK-3/4 S2-32APSK-4/5 S2-32APSK-5/6 S2-32APSK-8/9 S2-32APSK-9/10 S2X-QPSK-13/45 S2X-QPSK-9/20 S2X-QPSK-11/20 S2X-8APSK-5/9-L S2X-8APSK-26/45-L S2X-8PSK-23/36 S2X-8PSK-25/36 S2X-8PSK-13/18 S2X-16APSK-1/2-L S2X-16APSK-8/15-L S2X-16APSK-5/9-L S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-3/5-L S2X-16APSK-28/45 S2X-16APSK-23/36 S2X-16APSK-2/3-L S2X-16APSK-25/36 S2X-16APSK-13/18 S2X-16APSK-7/9 S2X-16APSK-77/90 S2X-32APSK-2/3-L S2X-32APSK-32/45 S2X-32APSK-11/15 S2X-32APSK-7/9 S2X-64APSK-32/45-L S2X-64APSK-11/15 S2X-64APSK-7/9 S2X-64APSK-6/5 S2X-64APSK-5/6 S2X-128APSK-3/4 S2X-128APSK-7/9 S2X-256APSK-29/45-L S2X-256APSK-2/3-L S2X-256APSK-31/45- S2X-256APSK-32/45 S2X-256APSK-11/15-L S2X-256APSK-3/4 S2X-QPSK-11/45 S2X-QPSK-4/15 S2X-QPSK-14/45 S2X-QPSK-7/15 S2X-QPSK-8/15 S2X-QPSK-32/45 S2X-8PSK-7/15 S2X-8PSK-8/15 S2X-8PSK-26/45 S2X-8PSK-32/45 S2X-16APSK-7/15 S2X-16APSK-8/15 S2X-16APSK-26/45 S2X-16APSK-3/5 S2X-16APSK-32/45 S2X-32APSK-2/3 S2X-32APSK-32/45)
tx.mts.1.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mts.2.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mts.3.standard	CHOICE		EnumRange (DVB-S DVB-S2)

name	*type*	*flags*	*range*
tx.mts.4.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mts.5.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mts.6.standard	CHOICE		EnumRange (DVB-S DVB-S2)
tx.mts.1.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.2.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.3.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.4.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.5.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.6.modulation	CHOICE		EnumRange (BPSK QPSK 8PSK 8APSK 16QAM 16APSK 32APSK 64APSK 128APSK 256APSK N/A)
tx.mts.1.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.2.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)

name	*type*	*flags*	*range*
tx.mts.3.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.4.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.5.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.6.fec	CHOICE		EnumRange (1/2 2/3 3/4 5/6 6/7 7/8 8/9 9/10 1/4 1/3 2/5 3/5 4/5 11/15 11/20 11/45 13/18 13/45 14/45 23/36 25/36 26/45 28/45 31/45 32/45 4/15 5/9 7/15 7/9 77/90 8/15 9/20 N/A)
tx.mts.1.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.2.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.3.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.4.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.5.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.6.pilots	CHOICE		EnumRange (OFF ON GLOBAL)
tx.mts.1.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.2.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.3.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.4.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.5.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)

name	*type*	*flags*	*range*
tx.mts.6.fecShortFrame	CHOICE		EnumRange (SHORT NORMAL)
tx.mts.1.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.2.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.3.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.4.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.5.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.6.bbMuxInput	CHOICE		EnumRange (OFF ON)
tx.mts.1.upl	TEXT		StringRange
tx.mts.2.upl	TEXT		StringRange
tx.mts.3.upl	TEXT		StringRange
tx.mts.4.upl	TEXT		StringRange
tx.mts.5.upl	TEXT		StringRange
tx.mts.6.upl	TEXT		StringRange
tx.mts.1.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.2.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.3.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.4.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.5.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.6.mode	CHOICE		EnumRange (VCM/ACM CCM)
tx.mts.1.tsgs	TEXT		StringRange
tx.mts.2.tsgs	TEXT		StringRange
tx.mts.3.tsgs	TEXT		StringRange
tx.mts.4.tsgs	TEXT		StringRange
tx.mts.5.tsgs	TEXT		StringRange
tx.mts.6.tsgs	TEXT		StringRange
tx.mts.1.generic	TEXT		StringRange

name	*type*	*flags*	*range*
tx.mts.2.generic	TEXT		StringRange
tx.mts.3.generic	TEXT		StringRange
tx.mts.4.generic	TEXT		StringRange
tx.mts.5.generic	TEXT		StringRange
tx.mts.6.generic	TEXT		StringRange
tx.mts.1.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.2.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.3.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.4.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.5.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.6.stillPicture	CHOICE		EnumRange (OFF ON)
tx.mts.1.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.2.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.3.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.4.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.5.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.6.faultMode	CHOICE		EnumRange (ELECTRICAL FLT TS FLT UP FLT)
tx.mts.1.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.2.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.3.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.4.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.5.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)
tx.mts.6.faultTime	FLOAT		DoubleRange (0.0 .. 0.0)

name	*type*	*flags*	*range*
tx.mts.1.mergerFullThr	TEXT		StringRange
tx.mts.2.mergerFullThr	TEXT		StringRange
tx.mts.3.mergerFullThr	TEXT		StringRange
tx.mts.4.mergerFullThr	TEXT		StringRange
tx.mts.5.mergerFullThr	TEXT		StringRange
tx.mts.6.mergerFullThr	TEXT		StringRange
tx.mts.1.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.2.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.3.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.4.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.5.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.6.peakPeriod	INTEGER		IntegerRange (0 .. 1937066666)
tx.mts.1.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.2.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.3.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.4.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.5.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.6.limiter	CHOICE		EnumRange (OFF ON)
tx.mts.1.framingMode	CHOICE		EnumRange (188 204)
tx.mts.2.framingMode	CHOICE		EnumRange (188 204)
tx.mts.3.framingMode	CHOICE		EnumRange (188 204)
tx.mts.4.framingMode	CHOICE		EnumRange (188 204)
tx.mts.5.framingMode	CHOICE		EnumRange (188 204)
tx.mts.6.framingMode	CHOICE		EnumRange (188 204)
tx.mts.1.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.2.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.3.syncByte	CHOICE		EnumRange (0xB8 0x47)

name	*type*	*flags*	*range*
tx.mts.4.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.5.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.6.syncByte	CHOICE		EnumRange (0xB8 0x47)
tx.mts.1.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.2.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.3.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.4.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.5.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.6.autoPolarity	CHOICE		EnumRange (OFF ON)
tx.mts.1.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.2.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.3.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.4.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.5.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.6.autoEqual	CHOICE		EnumRange (OFF ON)
tx.mts.1.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.2.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.3.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.4.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.5.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.6.tsoip	CHOICE		EnumRange (ASI TSOIP)
tx.mts.1.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.2.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.3.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.4.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.5.red.protection	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.mts.6.red.protection	CHOICE		EnumRange (OFF ON)
tx.mts.1.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.2.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.3.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.4.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.5.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.6.red.mode	CHOICE		EnumRange (AUTO MANUAL)
tx.mts.1.red.location	CHOICE		EnumRange (1 2)
tx.mts.2.red.location	CHOICE		EnumRange (1 2)
tx.mts.3.red.location	CHOICE		EnumRange (3 4)
tx.mts.4.red.location	CHOICE		EnumRange (3 4)
tx.mts.5.red.location	CHOICE		EnumRange (5 6)
tx.mts.6.red.location	CHOICE		EnumRange (5 6)
tx.mts.1.actualAsi	CHOICE	R/O	EnumRange R/O (1 2)
tx.mts.2.actualAsi	CHOICE	R/O	EnumRange R/O (1 2)
tx.mts.3.actualAsi	CHOICE	R/O	EnumRange R/O (3 4)
tx.mts.4.actualAsi	CHOICE	R/O	EnumRange R/O (3 4)
tx.mts.5.actualAsi	CHOICE	R/O	EnumRange R/O (5 6)
tx.mts.6.actualAsi	CHOICE	R/O	EnumRange R/O (5 6)
flags.mts.1.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mts.2.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mts.3.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mts.4.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mts.5.input	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.mts.6.input	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.si	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mz	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.a	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.b	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.aca	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asi.acb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mhx	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.mhu	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.asiIn	TEXT	R/O	StringRange R/O
internal.asiAuto	TEXT	R/O	StringRange R/O
internal.SpiIn	TEXT	R/O	StringRange R/O
internal.tx.on	TEXT	R/O	StringRange R/O
internal.fwOption	TEXT	R/O	StringRange R/O
internal.fwVersion	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
internal.fwMin	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fw	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.type	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Interpolator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Clock PLL overload)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Security device)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (FIFO full)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Clock oscillator)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Local oscillator)
faults.08	ALARM	R/O	AlarmFlagRange R/O (PLL lock)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Converter)
faults.10	ALARM	R/O	AlarmFlagRange R/O (UPC reference)
faults.11	ALARM	R/O	AlarmFlagRange R/O (UPC communication)
faults.12	ALARM	R/O	AlarmFlagRange R/O (DVB communication)
faults.13	ALARM	R/O	AlarmFlagRange R/O (TS clock osc.)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Transmit)
faults.15	ALARM	R/O	AlarmFlagRange R/O (External mute)
faults.16	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.17	ALARM	R/O	AlarmFlagRange R/O (DVB input sync)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RAM error)
faults.19	ALARM	R/O	AlarmFlagRange R/O (PCR memory)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Configuration)
faults.21	ALARM	R/O	AlarmFlagRange R/O (ASI/TSoIP A missing)
faults.22	ALARM	R/O	AlarmFlagRange R/O (ASI/TSoIP B missing)

9.781 Work-VSSBD

This page describes the device driver and the device window for the Work Microwave Synthesized dual block downconverters for new V-series.

The driver was tested with model:

- SSBD-Ka22

Device window pages

The following table shows which device window pages are available with this individual device type.

- RX --- The receive parameters
- --- The meter readings.
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the downconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)

name	*type*	*flags*	*range*
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
rx.on	CHOICE		EnumRange (OFF ON)
rx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
rx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
rx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
rx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (RX Off)

9.782 Work-VSSBU

This page describes the device driver and the device window for the Work Microwave Synthesized dual band block upconverters for new V-series.

Device window pages

The following table shows which device window pages are available with this individual device type.

- TX --- The transmit parameters
- --- The meter readings
- --- The redundancy switching parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the upconverter.

parameter	***description***
address	The device address (decimal) as set at the device's front panel.
Serial	Remember to set the line parameters for the serial interface.
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
extMuteInput	Defines to disable or enable the input for Hardware Mute (RF-Inhibit).
extRef	Defines the reference source.
autoSave	Store current parameters automatically in the instrument startup configuration.
bandSelect	Defines the used frequency band.
accessMode	**Full Control:** Enables changes on all rx parameters**Blocked by RSCM:** Disables changes on all rx parameters (readonly).
hasEqualizer	Defines if the unit is equipped with an equalizer.

Remote Control

Default TCP Port= 50505

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.FrontBaseVersion	TEXT	R/O	StringRange R/O
info.FrontDeviceVersion	TEXT	R/O	StringRange R/O
info.InterfaceVersion	TEXT	R/O	StringRange R/O
info.ReferenceVersion	TEXT	R/O	StringRange R/O
info.SynthesizerVersion	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.extMuteInput	CHOICE	SETUP	EnumRange (ENABLED DISABLED)
config.extRef	CHOICE	SETUP	EnumRange (INTERNAL EXTERNAL AUTO)
config.autoSave	CHOICE	SETUP	EnumRange (OFF ON)
config.bandSelect	CHOICE	SETUP	EnumRange (BAND-1 BAND-2)
config.accessMode	CHOICE	SAVE SETUP	EnumRange (FULL CONTROL BLOCKED BY RSCM)
config.hasEqualizer	CHOICE	SAVE SETUP	EnumRange (NO YES)

name	*type*	*flags*	*range*
config.attnRange	CHOICE	SAVE SETUP	EnumRange (20 30)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.frequency	FLOAT		DoubleRange (0.000 .. 0.000)
tx.eq	FLOAT		DoubleRange DIS (0.0 .. 8.0)
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.extRef	BOOLEAN	R/O	BooleanRange R/O (true false)
meas.extMute	BOOLEAN	R/O	BooleanRange R/O (true false)
reset	TEXT		StringRange
internal.sa0	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (OCXO oven cold)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Internal)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Reference)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Amplifier)
faults.06	ALARM	R/O	AlarmFlagRange R/O (MW output)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Synthesizer)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Block synthesizer)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External Ref)
faults.26	ALARM	R/O	AlarmFlagRange R/O (TX Off)

9.783 XMW-Blockupconverter

This page describes the device driver and the device window for the XMW Blockupconverter Series using XMWs own UDP Packet protocol. The driver has been developed and tested with 1W KaBand BUC T28E01, T29E01, T30E01 outdoor blockupconverter using SW version 1.1.

Communication port

Default Port for UDP communication used is 4001 if not otherwise specified. IP address and communication ports are defined in device setups address field.

- IpAddress writes to port 4001, reads from port 4001
- IpAddress:4002 writes to port 4002, reads from 4002
- IpAddress:4001:4002 writes to port 4001, reads from 4002

Please note: If you have more than one BUC in your setup please use individual ports to distinguish the device responses

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- TX --- The operational parameters (RF ON/OFF)
- --- Displays meter readings.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the BUC.

parameter	***description***
address	The device IP address and communication ports. See section Communication Port
readBackDelay	Defines the delay the driver uses when it reads back parameters recently commanded. The delay is initialized at a proved preset value which should not be changed under normal circumstances.
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.signal.on	CHOICE	R/O	EnumRange R/O (OFF ON)
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
config.deviceComment	TEXT	SAVE SETUP	StringRange
meas.temperature	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.voltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
tx.on	CHOICE		EnumRange (ON OFF)
faults.01	ALARM	R/O	AlarmFlagRange R/O (PLL Unlock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Over Temperature)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Over Power)
faults.04	ALARM	R/O	AlarmFlagRange R/O (External Mute)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
internal.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.784 Xicom-KHPA

This page describes the device driver and the device window for the Xicom Klystron amplifiers.

The driver has been tested with a XTK-3000C amplifier.

This driver has no support for fast tuner and redundancy.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 65
lowLimCheck	Configures if the amplifier shall check a low output level threshold.
lowLimit	The threshold value (Watts) to be applied if the setting above is activated.
highLimCheck	Configures if the amplifier shall check a high output level threshold.
highLimit	The threshold value (Watts) to be applied if the setting above is activated.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.tubeId	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY BEAM ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 30.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 30.0)
tx.requestedLevel	FLOAT		DoubleRange (-100.0 .. 40.0)
tx.autoHeater	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
tx.powerSave	CHOICE		EnumRange (OFF ON)
tx.constantLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
tx.constantPower	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.refPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iBeam	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uBeam	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iBody	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.drive	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.psTemp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.blowPres	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.blowFreq	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.x3	TEXT	R/O	StringRange R/O
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvpending	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.constpwr	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.heater	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pwrsave	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pout	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.pref	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pfwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Blower)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.07	ALARM	R/O	AlarmFlagRange R/O (KPA temp)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Body current)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Latch Body current)
faults.10	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.11	ALARM	R/O	AlarmFlagRange R/O (High output pwr)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Low output pwr)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Ext. interlock)
faults.14	ALARM	R/O	AlarmFlagRange R/O (High refl. pwr)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.785 Xicom-TWTA-XT

This page describes the device driver and the device window for the Xicom TWT XT Series amplifiers. The driver has been tested with a XT-100 without attenuator via XTC-100D single HPA controller and with old firmware 3.28.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters and meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*
address	The device address (decimal). The factory default is 65
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (OFF ON)
reset	TEXT	NOPRESET	StringRange
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pson	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.autop	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fan locked)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Momentary helix arc)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched helix arc)
faults.09	ALARM	R/O	AlarmFlagRange R/O (High reflected pwr)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.14	ALARM	R/O	AlarmFlagRange R/O (TX Inhibited)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.786 Xicom-TWTA-XTD

This page describes the device driver and the device window for the Xicom TWT XTD Series amplifiers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters and meter readings
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 65
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.maxAttenuation	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 100.0)
config.lowLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.redundancy	CHOICE	SETUP	EnumRange (STANDALONE 1:1 REDUNDANT 1+1 REDUNDANT 1:1 EXTRA WGS 2:1 REDUNDANT)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY BEAM ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
state	TEXT	R/O	StringRange R/O
tx.constantLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)

name	*type*	*flags*	*range*
tx.constantPower	CHOICE		EnumRange (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.iHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
red.switch	CHOICE		EnumRange DIS (ANTENNA LOAD)
red.mode	CHOICE		EnumRange DIS (AUTOMATIC MANUAL)
red.switch2	CHOICE	R/O	EnumRange R/O DIS (ANTENNA LOAD)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.x3	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.pref	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pfwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sc	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.sd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.aa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ab	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.constpwr	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvpending	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.automatic	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1ant	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1load	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2A	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2B	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)

name	*type*	*flags*	*range*
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Momentary helix arc)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched helix arc)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Reflected power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Overdrive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF power limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Fast VSWR)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Too much drive pwr)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Const pwr suspended)

name	*type*	*flags*	*range*
faults.24	ALARM	R/O	AlarmFlagRange R/O (Insufficient drive pwr)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External inhibit)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.29	ALARM	R/O	AlarmFlagRange R/O (Phase Lock)
faults.30	ALARM	R/O	AlarmFlagRange R/O (External Reference)
faults.31	ALARM	R/O	AlarmFlagRange R/O (Heater)
faults.32	ALARM	R/O	AlarmFlagRange R/O (Fan)

9.787 Xicom-TWTA-XTDL

This page describes the device driver and the device window for the Xicom TWT XTDL series amplifiers. The driver has been tested with the following models:

- XTD-500Ka outdoor amplifier SW release 0.07

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the amplifier.

parameter	*description*
address	The device address (decimal). The factory default is 65
powerDisplayMode	Defines the display of gain or attenuation in the parameter section.
maxAttenuation	Defines the maximum possible attenuation level as it varies and could not be read automatically from the device.
lowLimCheck	Configures if the amplifier shall check a low output level threshold.
lowLimit	The threshold value (Watts) to be applied if the setting above is activated.
highLimCheck	Configures if the amplifier shall check a high output level threshold.
highLimit	The threshold value (Watts) to be applied if the setting above is activated.
redundancy	Configures the operation mode of the internal redundancy switching logic if the amplifier is used in a redundant setup which is controlled by the amplifiers or a Xicom controller.
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.maxAttenuation	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 100.0)
config.lowLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.redundancy	CHOICE	SETUP	EnumRange (STANDALONE 1:1 REDUNDANT 1+1 REDUNDANT 1:1 EXTRA WGS 2:1 REDUNDANT)
config.hasUPC	CHOICE	SAVE SETUP	EnumRange (NO YES)
config.upcVoltageRange	CHOICE	SETUP	EnumRange (0..10V -5..5V -10..0V)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY BEAM ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
state	TEXT	R/O	StringRange R/O
tx.constantLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
tx.constantPower	CHOICE		EnumRange (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
reset	TEXT	NOPRESET	StringRange
red.switch	CHOICE		EnumRange DIS (ANTENNA LOAD)
red.mode	CHOICE		EnumRange DIS (AUTOMATIC MANUAL)
red.switch2	CHOICE	R/O	EnumRange R/O DIS (ANTENNA LOAD)
upc.voltage	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
upc.enabled	CHOICE	R/O	EnumRange R/O DIS (N/A ON OFF)

name	*type*	*flags*	*range*
upc.enable	CHOICE		EnumRange DIS (OFF ON)
upc.clearSkyAttn	FLOAT		DoubleRange DIS (0.0 .. 38.0)
upc.clearSkyVoltage	FLOAT		DoubleRange DIS (0.0 .. 10.0)
upc.boost	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.txFade	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.fade	FLOAT	R/O	DoubleRange R/O DIS (0.0 .. 0.0)
upc.slope	FLOAT		DoubleRange DIS (-2.00 .. 2.00)
upc.manualRatio	FLOAT		DoubleRange DIS (0.1 .. 9.9)
upc.hysteresis	FLOAT		DoubleRange DIS (0.0 .. 10.0)
upc.clearSkyWidth	FLOAT		DoubleRange DIS (0.0 .. 10.0)
upc.clearSkyCeiling	CHOICE		EnumRange DIS (OFF ON)
upc.boostLimit	FLOAT		DoubleRange DIS (0.0 .. 20.0)
upc.maxPower	FLOAT		DoubleRange DIS (23.0 .. 70.0)
upc.interval	FLOAT		DoubleRange DIS (0.10 .. 15.00)
upc.maxStep	FLOAT		DoubleRange DIS (0.1 .. 5.0)
upc.offset	FLOAT		DoubleRange DIS (-3.0 .. 3.0)
upc.flags.power	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
upc.flags.attn	BOOLEAN	R/O	BooleanRange R/O DIS (true false)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.x21	TEXT	R/O	StringRange R/O
internal.x3	TEXT	R/O	StringRange R/O
internal.x31	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.pref	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pfwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)

name	*type*	*flags*	*range*
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.aa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ab	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.constpwr	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvpending	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.automatic	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1ant	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1load	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2A	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2B	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.upc	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.faults.84	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fan locked)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Momentary helix arc)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched helix arc)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Reflected power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF power limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Upconverter)

name	*type*	*flags*	*range*
faults.21	ALARM	R/O	AlarmFlagRange R/O (Fast VSWR)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Too much drive pwr)
faults.23	ALARM	R/O	AlarmFlagRange R/O (Const pwr suspended)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Insufficient drive pwr)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External inhibit)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.81	ALARM	R/O	AlarmFlagRange R/O (UPC Power Limit)
faults.83	ALARM	R/O	AlarmFlagRange R/O (UPC Attn Limit)
faults.84	ALARM	R/O	AlarmFlagRange R/O (UPC suspended)
faults.126	ALARM	R/O	AlarmFlagRange R/O (Beam/HV Off)

9.788 Xicom-TWTA-XTRT

This page describes the device driver and the device window for the Xicom TWT XTRT Series amplifiers.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters and meter readings
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	***description***
address	The device address (decimal). The factory default is 65
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.version	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
firstRun	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.maxAttenuation	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 100.0)
config.lowLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.redundancy	CHOICE	SETUP	EnumRange (STANDALONE 1:1 REDUNDANT 1+1 REDUNDANT 1:1 EXTRA WGS 2:1 REDUNDANT)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY BEAM ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 0.0)
tx.attn	FLOAT		DoubleRange (0.0 .. 0.0)
tx.requestedLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
state	TEXT	R/O	StringRange R/O
minEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)
maxEirp	FLOAT	SAVE	DoubleRange (0.00 .. 0.00)

name	*type*	*flags*	*range*
eirp	FLOAT		DoubleRange (0.00 .. 0.00)
tx.constantLevel	FLOAT	SAVE	DoubleRange (-100.0 .. 40.0)
tx.constantPower	CHOICE		EnumRange (OFF ON)
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.temp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.uHeater	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
reset	TEXT	NOPRESET	StringRange
red.switch	CHOICE		EnumRange DIS (ANTENNA LOAD)
red.mode	CHOICE		EnumRange DIS (AUTOMATIC MANUAL)
red.switch2	CHOICE	R/O	EnumRange R/O DIS (ANTENNA LOAD)
position	CHOICE		EnumRange (3-STBY 1-STBY 2-STBY)
position.1	CHOICE	R/O	EnumRange R/O (A B)
position.2	CHOICE	R/O	EnumRange R/O (A B)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.x3	TEXT	R/O	StringRange R/O
internal.pout	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.pref	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pfwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.sc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.qd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.aa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ab	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ac	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.constpwr	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvpending	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.automatic	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1ant	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1load	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2A	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2B	BOOLEAN	R/O	BooleanRange R/O (true false)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)

name	*type*	*flags*	*range*
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Momentary helix arc)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched helix arc)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External interlock)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Reflected power)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Thermal interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Overdrive)
faults.19	ALARM	R/O	AlarmFlagRange R/O (RF power limit)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Fast VSWR)
faults.22	ALARM	R/O	AlarmFlagRange R/O (Too much drive pwr)

name	*type*	*flags*	*range*
faults.23	ALARM	R/O	AlarmFlagRange R/O (Const pwr suspended)
faults.24	ALARM	R/O	AlarmFlagRange R/O (Insufficient drive pwr)
faults.25	ALARM	R/O	AlarmFlagRange R/O (External inhibit)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)
faults.27	ALARM	R/O	AlarmFlagRange R/O (Protection)
faults.28	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)

9.789 Xicom-TWTA

This page describes the device driver and the device window for the Xicom TWT amplifiers. The driver has been tested with a XTD-200K outdoor amplifier.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters (BEAM-ON/STANDBY, RF ON/OFF)
- --- Displays meter readings like output power and TWT specific measurements.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*
address	The device address (decimal). The factory default is 65
lowLimCheck	Configures if the amplifier shall check a low output level threshold.
lowLimit	The threshold value (Watts) to be applied if the setting above is activated.

parameter	*description*
highLimCheck	Configures if the amplifier shall check a high output level threshold.
highLimit	The threshold value (Watts) to be applied if the setting above is activated.
redundancy	Configures the operation mode of the internal redundancy switching logic of the amplifier.
redundancyCheck	set to autodetect if you model is able to report its own redundancy configuration or to manual if not (needed by older firmware versions if you get an communication fault).
Serial	Remember to set the line parameters for the serial interface connected to the TWTA. The factory default of the HPA is *9600,N,8,1* .
logGainChanges	Defines if gain changes should be written in the event log database. Can automatically be adjusted by *logLevelEnabledId* within Uplink Power Control
helixAdvWarning	Sets the advanced helix current warning level. Not monitored if 0
deviceComment	An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.revision	TEXT	R/O	StringRange R/O
info.date	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.heatElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.beamElapsed	INTEGER	R/O	IntegerRange R/O (0 .. 100000)
info.signal.on	BOOLEAN	R/O	BooleanRange R/O (true false)
config.powerDisplayMode	CHOICE	SAVE SETUP	EnumRange (GAIN ATTN)
config.lowLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.lowLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.highLimCheck	CHOICE	SETUP	EnumRange (DISABLED FAULT ALARM)
config.highLimit	INTEGER	SETUP	IntegerRange (0 .. 0)
config.redundancy	CHOICE	SETUP	EnumRange (STANDALONE 1:1 REDUNDANT 1+1 REDUNDANT 1:1 EXTRA WGS 2:1 REDUNDANT)
config.redundancyCheck	CHOICE	SAVE SETUP	EnumRange (AUTODETECT MANUAL)
config.helixAdvWarning	FLOAT	SAVE SETUP	DoubleRange (0.0 .. 0.0)
config.logGainChanges	CHOICE	SAVE SETUP	EnumRange (ON OFF)
config.deviceComment	TEXT	SAVE SETUP	StringRange
tx.on	CHOICE		EnumRange (OFF ON)
tx.beam	CHOICE		EnumRange (STANDBY BEAM ON)
tx.gain	FLOAT		DoubleRange (0.0 .. 26.8)
tx.attn	FLOAT		DoubleRange (0.0 .. 26.8)
reset	TEXT	NOPRESET	StringRange

name	*type*	*flags*	*range*
state	TEXT	R/O	StringRange R/O
tx.measuredLevel	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
tx.measuredLevel2	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.fwdPwr	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.refPwr	INTEGER	R/O	IntegerRange R/O (0 .. 0)
meas.iHelix	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
meas.uHelix	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
meas.temp	FLOAT	R/O	DoubleRange R/O (0 .. 0)
red.switch	CHOICE		EnumRange DIS (ANTENNA LOAD)
red.mode	CHOICE		EnumRange DIS (AUTOMATIC MANUAL)
red.switch2	CHOICE	R/O	EnumRange R/O DIS (ANTENNA LOAD)
internal.x1	TEXT	R/O	StringRange R/O
internal.x2	TEXT	R/O	StringRange R/O
internal.x3	TEXT	R/O	StringRange R/O
internal.sa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.inhibit	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.standby	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvon	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hvpending	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.remote	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.automatic	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw1ant	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
internal.sw1load	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2A	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.sw2B	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.pout	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.pref	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pfwd	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
internal.fa	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fb	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fc	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.fd	INTEGER	R/O	IntegerRange R/O (0 .. 0)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Remote access)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Low line)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Fan locked)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Over voltage)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Over temp)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Momentary helix arc)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Latched helix arc)
faults.09	ALARM	R/O	AlarmFlagRange R/O (Waveguide arc)
faults.10	ALARM	R/O	AlarmFlagRange R/O (High RF)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Low RF)
faults.12	ALARM	R/O	AlarmFlagRange R/O (External interlock)

name	*type*	*flags*	*range*
faults.13	ALARM	R/O	AlarmFlagRange R/O (VSWR)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Under voltage)
faults.15	ALARM	R/O	AlarmFlagRange R/O (Cover interlock)
faults.16	ALARM	R/O	AlarmFlagRange R/O (Thermal Interlock)
faults.17	ALARM	R/O	AlarmFlagRange R/O (Overdrive)
faults.18	ALARM	R/O	AlarmFlagRange R/O (RF deck over temp)
faults.19	ALARM	R/O	AlarmFlagRange R/O (Overdrive trip)
faults.20	ALARM	R/O	AlarmFlagRange R/O (Upconverter)
faults.21	ALARM	R/O	AlarmFlagRange R/O (Helix advance warning)
faults.26	ALARM	R/O	AlarmFlagRange R/O (Standby)

9.790 Xytronix-X600

This page describes the device driver and the device window for the*Xytronix X-400 / X-600* I/O controller family. The device driver is actually limited to monitor and control a X-400 or X-600 main module with one to six X-11S dual relay output modules.

The state of the relays are displayed as ON/OFF at drivers device window, the may be controlled there directly. In parallel, the driver provides an integer variable for each relay using the values 0/1 for the relay states. These integer parameters may be used to integrate the relays into the sat-nms framework for M&C of plain digital I/O functions, permitting to access each relay through an Output logical device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows / commands the relay states.
- --- The faults page.
- --- The device info page.

- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

parameter	***description***
address	Must be set to the IP address of the X-400 / X-600 module
relays	Must be set to the number of relays actually controlled by the X-400 / X-600 module. The relays must be enumerated consecutively in X-400 / X-600 Web-UI, starting at 'relay1'.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.supplyVoltage	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
info.time	TEXT	R/O	StringRange R/O
config.relays	CHOICE	SAVE SETUP	EnumRange (2 4 6 8 10 12 14 16)
config.polarity	CHOICE	SAVE SETUP	EnumRange (NRM INV)
relay.num.1	INTEGER		IntegerRange (0 .. 1)
relay.num.2	INTEGER		IntegerRange (0 .. 1)
relay.num.3	INTEGER		IntegerRange (0 .. 1)
relay.num.4	INTEGER		IntegerRange (0 .. 1)

name	*type*	*flags*	*range*
relay.num.5	INTEGER		IntegerRange (0 .. 1)
relay.num.6	INTEGER		IntegerRange (0 .. 1)
relay.num.7	INTEGER		IntegerRange (0 .. 1)
relay.num.8	INTEGER		IntegerRange (0 .. 1)
relay.num.9	INTEGER		IntegerRange (0 .. 1)
relay.num.10	INTEGER		IntegerRange (0 .. 1)
relay.num.11	INTEGER		IntegerRange (0 .. 1)
relay.num.12	INTEGER		IntegerRange (0 .. 1)
relay.num.13	INTEGER		IntegerRange (0 .. 1)
relay.num.14	INTEGER		IntegerRange (0 .. 1)
relay.num.15	INTEGER		IntegerRange (0 .. 1)
relay.num.16	INTEGER		IntegerRange (0 .. 1)
relay.on.1	CHOICE		EnumRange (OFF ON)
relay.on.2	CHOICE		EnumRange (OFF ON)
relay.on.3	CHOICE		EnumRange (OFF ON)
relay.on.4	CHOICE		EnumRange (OFF ON)
relay.on.5	CHOICE		EnumRange (OFF ON)
relay.on.6	CHOICE		EnumRange (OFF ON)
relay.on.7	CHOICE		EnumRange (OFF ON)
relay.on.8	CHOICE		EnumRange (OFF ON)
relay.on.9	CHOICE		EnumRange (OFF ON)
relay.on.10	CHOICE		EnumRange (OFF ON)
relay.on.11	CHOICE		EnumRange (OFF ON)
relay.on.12	CHOICE		EnumRange (OFF ON)
relay.on.13	CHOICE		EnumRange (OFF ON)
relay.on.14	CHOICE		EnumRange (OFF ON)
relay.on.15	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
relay.on.16	CHOICE		EnumRange (OFF ON)

9.791 Zodiac-Cortex-ACU

This page describes the device driver and the device window for the Zodiac (In-Snec) Cortex NT ACU. The driver is an read-only driver which simply reports the actual antenna pointing, the tracking level and pointing deviations for up to two DTRs configured with the ACU. The driver does not permit to change any settings at the ACU.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Shows the antenna poining and the monitored tracking parameters.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window the following configuration parameters which must be set to make the software talk to the ACU/DTR system.

parameter	*description*
address	The Zodiac Cortex ACU does not use this parameter.
secondDTR	Tells the driver if the ACU provides a second DTR.

Device addressing

The Zodiac Cortex NT ACU is controlled by means of a binary protocol running over a TCP/IP socket connection. Hence, the device is to be configured in the same way than devices controlled through a TCP/IP to serial adapter / terminal server. The default port used by the device is 3000. The port specification for a device with the IP address 10.0.0.1 would be "10.0.0.1:3000".

Variables defined by this device driver

name	*type*	*flags*	*range*

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.poslog	TEXT	R/O	StringRange R/O
config.secondDTR	CHOICE	SAVE SETUP	EnumRange (NOT AVAILABLE AVAILABLE)
config.secondTDC	CHOICE	SAVE SETUP	EnumRange (NOT AVAILABLE AVAILABLE)
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.f	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.az	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.el	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.pl	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.mode1	TEXT	R/O	StringRange R/O
state.modeAz	TEXT	R/O	StringRange R/O
state.modeEl	TEXT	R/O	StringRange R/O
state.modePl	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.autoPhase	TEXT	R/O	StringRange R/O
state.beacon.1	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.beacon.2	FLOAT	R/O	DoubleRange R/O DIS (0.00 .. 0.00)
state.error.az.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.error.el.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.error.az.2	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
state.error.el.2	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
state.frequency.1	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.frequency.2	FLOAT	R/O	DoubleRange R/O DIS (0.000 .. 0.000)
tdc.frequency.1	FLOAT		DoubleRange (0.000 .. 0.000)
tdc.frequency.2	FLOAT		DoubleRange DIS (0.000 .. 0.000)
tdc.on.1	CHOICE		EnumRange (OFF ON)
tdc.on.2	CHOICE		EnumRange DIS (OFF ON)
azimuth	FLOAT		DoubleRange (0.000 .. 0.000)
elevation	FLOAT		DoubleRange (0.000 .. 0.000)
polarization	FLOAT		DoubleRange (0.000 .. 0.000)
trackMode	CHOICE		EnumRange (STANDBY MONOPULSE STEP TRACK REMOTE CMD EPHEMERIS SCAN THEME SLEW MANUAL COMPUTE LIMIT OUT MEMORY STOW AUTO CALIBRATION UNSTOW DIRECT SLEW MANUAL STANDBY JOYSTICK AUTO TEST UNDEFINED)
cmdstat	TEXT	R/O	StringRange R/O
internal.cmd	TEXT	R/O	StringRange R/O
faults.01	ALARM	R/O	AlarmFlagRange R/O (ACU Major summary)
faults.02	ALARM	R/O	AlarmFlagRange R/O (ACU Minor summary)
internal.modeAz	TEXT	R/O	StringRange R/O
internal.modeEl	TEXT	R/O	StringRange R/O
internal.modePl	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.trackAz	TEXT	R/O	StringRange R/O
internal.trackEl	TEXT	R/O	StringRange R/O

9.792 Zodiac-Cortex-CRT-IFM

This page describes the *Zodiac-Cortex-CRT-IFM* device driver and its device window. The *Zodiac-Cortex-CRT-IFM* device driver uses the following tables: IF Modulator, IF Modulator Extension and Embedded Frequency Converter. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

If more than one modulator is used on the CRT several instances of this driver can be added within the same thread

Beside the IFM multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Modulator settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- module --- Defines the modulator module to be controlled (IFM-1, IFM-2 or IFM-3).
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.basic	TEXT	R/O	StringRange R/O
info.extended	TEXT	R/O	StringRange R/O
info.embFreqConv	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.efcModel	TEXT	R/O	StringRange R/O
info.bit31	BOOLEAN	R/O	BooleanRange R/O (true false)
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.module	CHOICE	SAVE SETUP	EnumRange (IFM-1 IFM-2 IFM-3)

name	*type*	*flags*	*range*
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.cmdInt	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.ifmTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.ifmxTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.hasIfm	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasIfmX	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasEfc	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.if	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.rf	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.freqHz	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.freq32Bit	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.freqOffset	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.match1	TEXT	R/O	StringRange R/O
internal.match2	TEXT	R/O	StringRange R/O
internal.bit31	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.opMode	TEXT	R/O	StringRange R/O
internal.model	HEX	R/O	HexRange R/O (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O
if.frequency	FLOAT		DoubleRange (66.000000 .. 74.000000)
deviation.tcu	INTEGER		IntegerRange (0 .. 500000)
deviation.rau	INTEGER		IntegerRange (0 .. 500000)

name	*type*	*flags*	*range*
deviation.tms1	INTEGER		IntegerRange (0 .. 500000)
modulationIndex.tcu	FLOAT		DoubleRange (0.00 .. 2.50)
modulationIndex.rau	FLOAT		DoubleRange (0.00 .. 2.50)
modulationIndex.tms1	FLOAT		DoubleRange (0.00 .. 2.50)
operatingMode	CHOICE		EnumRange (FM PM BPSK QPSK OQPSK AQPSK SOQPSK GMSK 8PSK DVBS2)
enable.aux	CHOICE		EnumRange (OFF ON)
enable.rau	CHOICE		EnumRange (OFF ON)
enable.tcu1	CHOICE		EnumRange (OFF ON)
enable.tcu2	CHOICE		EnumRange (OFF ON)
enable.tms1	CHOICE		EnumRange (OFF ON)
enable.tms2	CHOICE		EnumRange (OFF ON)
enable.tms3	CHOICE		EnumRange (OFF ON)
enable.mod	CHOICE		EnumRange (TCU RAU TMS TCU+RAU RAU+TMS TCU+TMS AUX AUX+TCU AUX+RAU AUX+TMS AUC+TCU+RAU AUX+RAU+TMS AUX+TCU+TMS NONE)
modulation	CHOICE		EnumRange (ON OFF)
if.on	CHOICE		EnumRange (ON OFF)
if.spectrumInvert	CHOICE		EnumRange (NRM INV)
if.power	FLOAT		DoubleRange (-80.00 .. 10.00)
sweep.mode	CHOICE		EnumRange (OFF ON AUTO AUTO+RZ RZ DOPPLER STEP)
doppler.enable	CHOICE		EnumRange (OFF ON ON-IFR-LOOPBACK)
sweep.rate	FLOAT		DoubleRange (-175000.000 .. 175000.000)
sweep.range	FLOAT		DoubleRange (1.000 .. 1000.000)

name	*type*	*flags*	*range*
sweep.offset	FLOAT		DoubleRange (-1000.000 .. 1000.000)
span	CHOICE		EnumRange (19.531kHz 39.062kHz 78.125kHz 156.25kHz 312.5kHz 625kHz 1250kHz 2500kHz 5000kHz 10MHz 20MHz 9.765kHz 4.882kHz 2.441kHz 1.220kHz 610.351Hz)
deviation.aux	INTEGER		IntegerRange (0 .. 500000)
modulationIndex.aux	FLOAT		DoubleRange (0.00 .. 2.50)
deviation.pcm	INTEGER		IntegerRange (0 .. 500000)
modulationIndex.pcm	FLOAT		DoubleRange (0.00 .. 2.50)
matchedFilter	CHOICE		EnumRange (NO RC RRC SOQPSK-MIL SOQPSK-A SOQPSK-B SOQPSK-TG)
rollOff	FLOAT		DoubleRange (0.1 .. 1.0)
modulationIndex.prog	CHOICE		EnumRange (ON OFF)
sweep.stepSize	INTEGER		IntegerRange (10 .. 100000)
loopback.factor	FLOAT		DoubleRange (0.01 .. 8.00)
sub.frequency	FLOAT		DoubleRange (0.010000 .. 1.200000)
loopback	CHOICE		EnumRange (IFR-1 IFR-2 IFR-3 IFR-4 IFR-5)
deviation.tms2	INTEGER		IntegerRange (0 .. 500000)
modulationIndex.tms2	FLOAT		DoubleRange (0.00 .. 2.50)
if.bandwidth	CHOICE		EnumRange (NONE 125kHz 250kHz 500kHz 1MHz 2MHz 4MHz CUSTOM-1 CUSTOM-2 CUSTOM-3 CUSTOM-4 CUSTOM-5 CUSTOM-6 CUSTOM-7 CUSTOM-8)
qpskEncoding	CHOICE		EnumRange (NO-SINGLE DIFFERENTIAL DUAL DUAL+DIFF)

name	*type*	*flags*	*range*
mapping	CHOICE		EnumRange (GRAY-1 GRAY-2 GRAY-3 GRAY-4 GRAY-5 GRAY-6 NATURAL-1 NATURAL-2 INV-GRAY-1 INV-GRAY-2 INV-GRAY-3 INV-GRAY-4 INV-GRAY-5 INV-GRAY-6 INV-NAT-1 INV-NAT-2)
ratioIQ	FLOAT		DoubleRange (-20.00 .. 20.00)
state.offset	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.deviation	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.modulationIndex	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.span	TEXT	R/O	StringRange R/O
state.if.on	TEXT	R/O	StringRange R/O
state.mode	TEXT	R/O	StringRange R/O
state.doppler	TEXT	R/O	StringRange R/O
state.compensation	FLOAT	R/O	DoubleRange R/O (0.0 .. 0.0)
state.selection	TEXT	R/O	StringRange R/O
state.aux	TEXT	R/O	StringRange R/O
state.rau	TEXT	R/O	StringRange R/O
state.tcu1	TEXT	R/O	StringRange R/O
state.tcu2	TEXT	R/O	StringRange R/O
state.tms1	TEXT	R/O	StringRange R/O
state.tms2	TEXT	R/O	StringRange R/O
state.tms3	TEXT	R/O	StringRange R/O
mod.pilots	CHOICE		EnumRange (OFF ON)
mod.fecFrameType	CHOICE		EnumRange (NORMAL SHORT)
mod.type	CHOICE		EnumRange (DUMMY-PLFRAME QPSK-1/4 QPSK-1/3 QPSK-2/5 QPSK-1/2 QPSK-3/5 QPSK-2/3 QPSK-3/4 QPSK-4/5 QPSK-5/6 QPSK-8/9 QPSK-9/10 8PSK-3/5 8PSK-2/3 8PSK-5/6 8PSK-8/9 8PSK-9/10)

name	*type*	*flags*	*range*
mod.modulation	CHOICE		EnumRange (N/A QPSK 8PSK)
mod.fec	CHOICE		EnumRange (N/A 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5 5/6 8/9 9/10)
mod.plScramble	CHOICE		EnumRange (OFF ON)
mod.padding	CHOICE		EnumRange (OFF ON)
mod.bbHeader	CHOICE		EnumRange (OFF ON)
mod.dummy	CHOICE		EnumRange (OFF ON)
mod.matype	INTEGER		IntegerRange (0 .. 0)
mod.upl	INTEGER		IntegerRange (0 .. 0)
mod.sync	INTEGER		IntegerRange (0 .. 0)
state.datarate	FLOAT		DoubleRange (0 .. 0)
rf.frequency	FLOAT		DoubleRange (930.000000 .. 2450.000000)
rf.convFreq	FLOAT		DoubleRange (930.000000 .. 2450.000000)
rf.bypass	CHOICE		EnumRange (OFF ON)
rf.on	CHOICE		EnumRange (ON OFF)
rf.power	FLOAT		DoubleRange (-80.00 .. 10.00)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.pll	BOOLEAN	R/O	BooleanRange R/O (true false)
state.reference	TEXT	R/O	StringRange R/O
state.calibration	TEXT	R/O	StringRange R/O
sendCalibration	TEXT	NOPRESET	StringRange
resetCalibration	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (H/W)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EFC Connection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EFC PLL)

name	*type*	*flags*	*range*
faults.05	ALARM	R/O	AlarmFlagRange R/O (EFC Com)
faults.06	ALARM	R/O	AlarmFlagRange R/O (EFC Configuration)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EFC Calibration)

9.793 Zodiac-Cortex-CRT-IFR

This page describes the *Zodiac-Cortex-CRT-IFR* device driver and its device window. The *Zodiac-Cortex-CRT-IFR* device driver uses the following tables: IF Receiver, IF Receiver Extension and Embedded Frequency Converter. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

If more than one receiver is used on the CRT several instances of this driver can be added within the same thread

Beside the IFR multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Receiver settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at

this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- module --- Defines the receiver module to be controlled (IFR-1, IFR-2, IFR-3, IFR-4 or IFR-5).
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.basic	TEXT	R/O	StringRange R/O
info.extended	TEXT	R/O	StringRange R/O
info.embFreqConv	TEXT	R/O	StringRange R/O
info.efcModel	TEXT	R/O	StringRange R/O
info.test	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.module	CHOICE	SAVE SETUP	EnumRange (IFR-1 IFR-2 IFR-3 IFR-4 IFR-5)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.ifrTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.ifrxTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.hasIfr	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasIfrX	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasEfc	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.model	HEX	R/O	HexRange R/O (0 .. 0)
internal.offset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.if	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.rf	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.freqHz	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.freq32Bit	FLOAT	R/O	DoubleRange R/O (0 .. 0)
internal.freqOffset	FLOAT	R/O	DoubleRange R/O (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O
if.frequency	FLOAT		DoubleRange (66.000000 .. 75.000000)
inputPort	CHOICE		EnumRange (NOMINAL ALTERNATE)
input	CHOICE		EnumRange (IF-1 IF-2 IF-3 AUX-1 AUX-2 IF-4 IF-5 IF-6)
if.bandwidth	CHOICE		EnumRange (10MHz 1MHz 100kHz 5MHz 3MHz 1.5MHz 500kHz 300kHz 150kHz 50kHz 30kHz 15kHz 10kHz 5kHz 13MHz NONE)

name	*type*	*flags*	*range*
operatingMode	CHOICE		EnumRange (AUTO MANUAL)
carrierAcq	CHOICE		EnumRange (NORMAL HIGH)
loopBandwidth	CHOICE		EnumRange (30Hz 100Hz 300Hz 1000Hz 3000Hz)
loopBandwidthCont	FLOAT		DoubleRange (0.0 .. 3000.0)
videoDemodulation	CHOICE		EnumRange (PM FM AM-NO-COHERENT AM-COHERENT BYPASS SPECTRAL)
carrierAcqRange	FLOAT		DoubleRange (1.000 .. 500.000)
carrierTrkRange	FLOAT		DoubleRange (1.000 .. 500.000)
afc	CHOICE		EnumRange (OFF 1ms 10ms 100ms 1s)
agcTimeConstant	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
discriminator	INTEGER		IntegerRange (0 .. 16)
doppler.enable	CHOICE		EnumRange (OFF ON)
doppler.flowId	HEX		HexRange (0 .. 0)
doppler.sampling	FLOAT		DoubleRange (0.100000 .. 10.000000)
doppler.measurements	INTEGER		IntegerRange (1 .. 1000)
span.flowId	HEX		HexRange (0 .. 0)
span	CHOICE		EnumRange (1.220kHz 2.440kHz 4.882kHz 9.764kHz 19.53kHz 39.062kHz 78.124kHz 156.25kHz 312.5kHz 625kHz 1250kHz 2.5MHz 5MHz 10MHz)
pcmDemodulation	CHOICE		EnumRange (OFF BPSK PCM/PM QPSK OQPSK AQPSK SPAN SOQPSK GMSK 8PSK DVB-S2)
predetectionFilter	INTEGER		IntegerRange (0 .. 256)
videoFilter	INTEGER		IntegerRange (0 .. 256)

name	*type*	*flags*	*range*
qpskDecoding	CHOICE		EnumRange (NO-SINGLE DIFFERENTIAL DUAL DUAL+DIFF)
mapping8psk	CHOICE		EnumRange (GRAY-1 GRAY-2 GRAY-3 GRAY-4 GRAY-5 GRAY-6 NATURAL-1 NATURAL-2 INV-GRAY-1 INV-GRAY-2 INV-GRAY-3 INV-GRAY-4 INV-GRAY-5 INV-GRAY-6 INV-NAT-1 INV-NAT-2)
swapIQ	CHOICE		EnumRange (NO SWAP)
ratioIQ	FLOAT		DoubleRange (-20.0 .. 20.0)
loopBandwidthAcq	FLOAT		DoubleRange (0.0 .. 3000.0)
state.offset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.sampling	TEXT	R/O	StringRange R/O
state.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.deviation	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.doppler	TEXT	R/O	StringRange R/O
state.storage	TEXT	R/O	StringRange R/O
state.span	TEXT	R/O	StringRange R/O
state.demodulation	TEXT	R/O	StringRange R/O
rf.frequency	FLOAT		DoubleRange (930.000000 .. 2450.000000)
rf.convFreq	FLOAT		DoubleRange (930.000000 .. 2450.000000)
rf.bypass	CHOICE		EnumRange (OFF ON)
rf.gain	INTEGER		IntegerRange (15 .. 30)
rf.agc	CHOICE		EnumRange (OFF ON)
temperature	INTEGER	R/O	IntegerRange R/O (0 .. 0)
flags.pll	BOOLEAN	R/O	BooleanRange R/O (true false)
state.reference	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.calibration	TEXT	R/O	StringRange R/O
acquisition	TEXT	NOPRESET	StringRange
doppler.start	TEXT	NOPRESET	StringRange
doppler.stop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (PLL)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Bit Synchronizer)
faults.03	ALARM	R/O	AlarmFlagRange R/O (EFC Connection)
faults.04	ALARM	R/O	AlarmFlagRange R/O (EFC PLL)
faults.05	ALARM	R/O	AlarmFlagRange R/O (EFC Com)
faults.06	ALARM	R/O	AlarmFlagRange R/O (EFC Configuration)
faults.07	ALARM	R/O	AlarmFlagRange R/O (EFC Calibration)

9.794 Zodiac-Cortex-CRT-PRJ

This page describes the *Zodiac-Cortex-CRT-PRJ* device driver and its device window. The *Zodiac-Cortex-CRT-PRJ* device driver uses the following tables: Project specific (mPower). It is based on the DTU 101701 remote protocol document v 1.3, dated Dec. 16., 2021

Beside the PRJ multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Project specific settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.word	HEX	R/O	HexRange R/O (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O
tmu.bss1	BOOLEAN	NOPRESET	BooleanRange (true false)
tmu.bss2	BOOLEAN	NOPRESET	BooleanRange (true false)
tmu.bss3	BOOLEAN	NOPRESET	BooleanRange (true false)
tmu.bss4	BOOLEAN	NOPRESET	BooleanRange (true false)
frame.sizeOffset	INTEGER		IntegerRange (0 .. 65536)
pause	INTEGER		IntegerRange (0 .. 3600)
bitrate.1	INTEGER		IntegerRange (1000 .. 250000)
bitrate.2	INTEGER		IntegerRange (1000 .. 250000)
frequency.1	FLOAT		DoubleRange (5.000 .. 2000.000)
frequency.2	FLOAT		DoubleRange (5.000 .. 2000.000)
sync.word	HEX		HexRange (0 .. 0)
state.bss1	TEXT	R/O	StringRange R/O
state.bss2	TEXT	R/O	StringRange R/O
state.bss3	TEXT	R/O	StringRange R/O
state.bss4	TEXT	R/O	StringRange R/O
frame.bss1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
frame.bss2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
frame.bss3	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
frame.bss4	INTEGER	R/O	IntegerRange R/O (0 .. 0)

9.795 Zodiac-Cortex-CRT-TCU

This page describes the *Zodiac-Cortex-CRT-TCU* device driver and its device window. The *Zodiac-Cortex-CRT-TCU* device driver uses the following tables: Telecommand Unit. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

If more than one telecommand unit is used on the CRT several instances of this driver can be added within the same thread

Beside the TCU multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Telecommand unit settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- module --- Defines the telecommand module to be controlled (TCU-1 or TCU-2).
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.bandwidth	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.module	CHOICE	SAVE SETUP	EnumRange (TCU-1 TCU-2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.tcuTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.hasTcu	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.preamble	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
internal.idle	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.day.c1	HEX	R/O	HexRange R/O (0 .. 0)
internal.day.c4h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.day.c4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hour	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hour.c1	HEX	R/O	HexRange R/O (0 .. 0)
internal.secs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.msecs.c1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.msecs.c2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.msecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nsecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tag	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
days	INTEGER		IntegerRange (0 .. 0)
hours	INTEGER		IntegerRange (0 .. 0)
minutes	INTEGER		IntegerRange (0 .. 0)
seconds	INTEGER		IntegerRange (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O
dataFlowId	HEX		HexRange (0 .. 0)
frequency	FLOAT		DoubleRange (0.040 .. 2000.000)
frequency.tone1	INTEGER		IntegerRange (100 .. 100000)
frequency.tone2	INTEGER		IntegerRange (100 .. 100000)
frequency.tone3	INTEGER		IntegerRange (100 .. 100000)
symbolRate	INTEGER		IntegerRange (10 .. 40000000)

name	*type*	*flags*	*range*
operatingMode	CHOICE		EnumRange (NORMAL SIMULATION)
demodulationCheck	CHOICE		EnumRange (OFF ON)
shiftKeyingMode	CHOICE		EnumRange (FSK-BLANKING FSK BPSK 702 BPSK+AM 601 BPSK-SQUARE)
tonePhaseOffset	INTEGER		IntegerRange (0 .. 360)
pcm.code	CHOICE		EnumRange (NRZ-L NRZ-M BP-L RZ NRZ-S BP-M BP-S)
idlePattern	HEX		HexRange (0 .. 0)
preambleLength	INTEGER		IntegerRange (0 .. 1677724)
extPreamble	INTEGER		IntegerRange (0 .. 0)
idlePatternLength	INTEGER		IntegerRange (0 .. 8192)
extIdlePattern	INTEGER		IntegerRange (0 .. 0)
rfSetTime	FLOAT		DoubleRange (0.00 .. 9999.99)
cmm1.clcw	CHOICE		EnumRange (DISABLED ENABLED)
cmm2.clcw	CHOICE		EnumRange (DISABLED ENABLED)
clcwCheck	CHOICE		EnumRange (TM-CH-A TM-CH-B TM-CH-C TM-CH-D TM-CH-E TM-CH-F TM-CH-G TM-CH-H)
cmm1.offset	INTEGER		IntegerRange (0 .. 2043)
cmm1.mask	HEX		HexRange (0 .. 0)
cmm1.expectedValue	HEX		HexRange (0 .. 0)
cmm2.offset	INTEGER		IntegerRange (0 .. 2043)
cmm2.mask	HEX		HexRange (0 .. 0)
cmm2.expectedValue	HEX		HexRange (0 .. 0)
rfFallTime	FLOAT		DoubleRange (0.00 .. 9999.99)

name	*type*	*flags*	*range*
demodCheck	CHOICE		EnumRange (TM-CH-A TM-CH-B TM-CH-C TM-CH-D TM-CH-E TM-CH-F TM-CH-G TM-CH-H)
priority	CHOICE		EnumRange (TC+RNG TC)
amplitudeRatio	FLOAT		DoubleRange (0.0 .. 1.0)
pcm.modSignal	CHOICE		EnumRange (INTERNAL EXTERNAL)
pcm.clock	CHOICE		EnumRange (INTERNAL EXTERNAL)
convolutionalEnc	CHOICE		EnumRange (NONE G2-INV G2)
convolutionalRate	CHOICE		EnumRange (1/2 1/3 2/3 3/4-CONV1 3/4-CONV2 5/6-CONV1 5/6-CONV1 7/8)
doppler.enable	CHOICE		EnumRange (OFF ON)
frameEncoding	CHOICE		EnumRange (NONE SCRAMBLING)
ldpc.rate	CHOICE		EnumRange (OFF 1/2 2/3 4/5 7/8)
ldpc.block	CHOICE		EnumRange (OFF 1024 4096 7136 16384)
time.tag	CHOICE	R/O	EnumRange R/O (CODE-0 CODE-1 CODE-2 CODE-3 CODE-4 CODE-5)
state	TEXT	R/O	StringRange R/O
state.commandsExpected	TEXT	R/O	StringRange R/O
state.waitAbsTime	TEXT	R/O	StringRange R/O
state.dateOfExecution	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.dateOfExecution2	TEXT	R/O	StringRange R/O
state.lockedOut	TEXT	R/O	StringRange R/O
state.doppler	TEXT	R/O	StringRange R/O
state.sampling	TEXT	R/O	StringRange R/O
state.dataCount	TEXT	R/O	StringRange R/O
state.execCount	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
state.activeTestpoint	TEXT	R/O	StringRange R/O
waitForAbsTime	TEXT	NOPRESET	StringRange
abortPending	TEXT	NOPRESET	StringRange
idle	TEXT	NOPRESET	StringRange
sTone	TEXT	NOPRESET	StringRange
resetCounter	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (External Clock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Configuration)

9.796 Zodiac-Cortex-CRT-TMS

This page describes the *Zodiac-Cortex-CRT-TMS* device driver and its device window. The *Zodiac-Cortex-CRT-TMS* device driver uses the following tables: Telemetry Simulator. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

If more than one telemetry simulator is used on the CRT several instances of this driver can be added within the same thread.

Beside the TMS multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device

type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Telemetry simulator settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- module --- Defines the telemetry simuator module to be controlled (TMS-1, or TMS-2).
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.basic	TEXT	R/O	StringRange R/O
info.bw	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)

name	*type*	*flags*	*range*
config.module	CHOICE	SAVE SETUP	EnumRange (TMS-1 TMS-2)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.tmsTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.hasTms	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.test	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O
dataFlowId	HEX		HexRange (0 .. 0)
frequency	FLOAT		DoubleRange (0.000 .. 0.000)
symbolRate	INTEGER		IntegerRange (0 .. 0)
operatingMode	CHOICE		EnumRange (LAN FILE PSEUDO-RANDOM REPLAY-LEGACY BERT BER-RPN REPLAY-SINGLE)
output	CHOICE		EnumRange (OFF ON)
file	INTEGER		IntegerRange (1 .. 8)
pcm.code	CHOICE		EnumRange (BP-L NRZ-L NRZ-M NRZ-S BP-M BP-S DBP-M DBP-S DM-M DM-S R-NRZ NRZ-L-V35 NRZ-M-V35)
formatLength	INTEGER		IntegerRange (1 .. 32768)
modulation	CHOICE		EnumRange (NONE BPSK PCM-PM PCM-FM BPSK-SQUARE)
pcm.data	CHOICE		EnumRange (NORMAL INVERTED)
berFlow	CHOICE		EnumRange (TM-CH-A TM-CH-B TM-CH-C TM-CH-D TM-CH-E TM-CH-F TM-CH-G TM-CH-H)

name	*type*	*flags*	*range*
encoding	CHOICE		EnumRange (NONE SCRAMBLING G2-INV G2 G2-INV-SCRAMBLING G2-SCRAMBLING)
encodingRate	CHOICE		EnumRange (1/2 1/3 2/3 3/4-CONV1 3/4-CONV2 5/6-CONV1 5/6-CONV1 7/8)
frame.check	CHOICE		EnumRange (OFF ON)
frame.coding	CHOICE		EnumRange (NONE RS232-BETA TURBO RS232-ALPHA LDPC TPC RS239-BETA RS239-ALPHA)
pcm.modSignal	CHOICE		EnumRange (INTERNAL EXTERNAL)
pcm.clock	CHOICE		EnumRange (INTERNAL EXTERNAL)
frequencyDeviation	INTEGER		IntegerRange (0 .. 300000)
modulationIndex	FLOAT		DoubleRange (0.00 .. 2.50)
doppler.enable	CHOICE		EnumRange (OFF ON)
prn	CHOICE		EnumRange (STANDARD 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23)
replay	INTEGER		IntegerRange (0 .. 0)
state.ber	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
format.tm1	HEX	R/O	HexRange R/O (0 .. 0)
format.tm2	HEX	R/O	HexRange R/O (0 .. 0)
format.tm3	HEX	R/O	HexRange R/O (0 .. 0)
format.tm4	HEX	R/O	HexRange R/O (0 .. 0)
format.tm5	HEX	R/O	HexRange R/O (0 .. 0)
format.tm6	HEX	R/O	HexRange R/O (0 .. 0)
format.tm7	HEX	R/O	HexRange R/O (0 .. 0)
format.tm8	HEX	R/O	HexRange R/O (0 .. 0)
format.id	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
state.bytesLan	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.doppler	TEXT	R/O	StringRange R/O
state.sampling	TEXT	R/O	StringRange R/O
state.berCount	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fileId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.fileProgress	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.filetoReplay	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.activeTestpoint	TEXT	R/O	StringRange R/O
resetBer	TEXT	NOPRESET	StringRange
injectError	TEXT	NOPRESET	StringRange
start	TEXT	NOPRESET	StringRange
stop	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Ext Clock)
faults.02	ALARM	R/O	AlarmFlagRange R/O (Overflow)
faults.03	ALARM	R/O	AlarmFlagRange R/O (File)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Create Configuration)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Encoding)

9.797 Zodiac-Cortex-CRT-TMU

This page describes the *Zodiac-Cortex-CRT-TMU* device driver and its device window. The *Zodiac-Cortex-CRT-TMU* device driver uses the following tables: Telemetry Unit and Telemetry Unit extension. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

If more than one telemetry unit is used on the CRT several instances of this driver can be added within the same thread

Beside the TMU multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT monitors and controls the base units features of the CRT
- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Telemetry unit settings and status.
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- module --- Defines the telemetry module to be controlled (TMU-A, TMU-B, TMU-C, TMU-D, TMU-E, TMU-F, TMU-G or TMU-H).
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.basic	TEXT	R/O	StringRange R/O
info.extended	TEXT	R/O	StringRange R/O
info.bandwidth	TEXT	R/O	StringRange R/O
info.lowBwExtended	TEXT	R/O	StringRange R/O
info.rsDecoding	TEXT	R/O	StringRange R/O
info.viterbi	TEXT	R/O	StringRange R/O
info.turbo	TEXT	R/O	StringRange R/O
info.viterbiAllRate	TEXT	R/O	StringRange R/O
info.descrambling	TEXT	R/O	StringRange R/O
info.ldpc	TEXT	R/O	StringRange R/O
info.ldpcSclicing	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
info.dvbs2	TEXT	R/O	StringRange R/O
info.deaf	TEXT	R/O	StringRange R/O
info.lfsr	TEXT	R/O	StringRange R/O
info.utoa	TEXT	R/O	StringRange R/O
info.tpc	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.module	CHOICE	SAVE SETUP	EnumRange (TMU-A TMU-B TMU-C TMU-D TMU-E TMU-F TMU-G TMU-H)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)
internal.cmd	TEXT	R/O	StringRange R/O
internal.tmuTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.tmuxTable	HEX	R/O	HexRange R/O (0 .. 0)
internal.hasTmu	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.hasTmuX	BOOLEAN	R/O	BooleanRange R/O (true false)
internal.word	HEX	R/O	HexRange R/O (0 .. 0)
internal.word64	HEX	R/O	HexRange R/O (0 .. 0)
internal.wordMask	HEX	R/O	HexRange R/O (0 .. 0)
internal.wordMask64	HEX	R/O	HexRange R/O (0 .. 0)
internal.wordSet	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.coderate	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.match1	TEXT	R/O	StringRange R/O
internal.match2	TEXT	R/O	StringRange R/O
internal.ebNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
cmdstat	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
dataFlowId	HEX		HexRange (0 .. 0)
frequency	FLOAT		DoubleRange (0.040 .. 2000.000)
symbolRate	INTEGER		IntegerRange (10 .. 40000000)
loopBw	CHOICE		EnumRange (NARROW WIDE ULTRAWIDE)
pcm.code	CHOICE		EnumRange (BP-L NRZ-L NRZ-M NRZ-S BP-M BP-S DBP-M DBP-S DM-M DM-S R-NRZ NRZ-L-V35 NRZ-M-V35)
sync.wordMask	HEX		HexRange (0 .. 0)
decoding	CHOICE		EnumRange (NONE DESCRAMBLING G2-INV G2 G2-INV-SCRAMBLING G2-SCRAMBLING)
decodingRate	CHOICE		EnumRange (1/2 1/3 2/3 3/4-CONV1 3/4-CONV2 5/6-CONV1 5/6-CONV1 7/8)
rsCoding	CHOICE		EnumRange (NONE RS232-BETA TURBO RS232-ALPHA LDPC TPC RS239-BETA RS239-ALPHA)
descrambling	CHOICE		EnumRange (DEFAULT CCSDS LFSR-BEFORE LFSR-AFTER LFSR)
frame.size	INTEGER		IntegerRange (10 .. 32000)
sync.word	HEX		HexRange (0 .. 0)
sync.wordLength	INTEGER		IntegerRange (8 .. 64)
operatingMode	CHOICE		EnumRange (NORMAL TEST-1 TEST-2)
sync.threshold	INTEGER		IntegerRange (0 .. 64)
ctl.threshold	INTEGER		IntegerRange (0 .. 7)
lts.threshold	INTEGER		IntegerRange (0 .. 7)
bitSlipWindow	INTEGER		IntegerRange (0 .. 2)
frame.synchronizer	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
afl	FLOAT		DoubleRange (-300000.000000 .. 300000.000000)
frame.check	CHOICE		EnumRange (OFF CRC CHECKSUM)
crc.polynominal	HEX		HexRange (0 .. 0)
crc.preset	HEX		HexRange (0 .. 0)
accumulation	CHOICE		EnumRange (OFF 8 16 32 64)
sync.word64	HEX		HexRange (0 .. 0)
sync.wordMask64	HEX		HexRange (0 .. 0)
loopBw.bitSync	FLOAT		DoubleRange (0.00 .. 1.00)
input	CHOICE		EnumRange (INT EXT-FS EXT-VIT EXT-VIT+BS IQ)
rollOff	FLOAT		DoubleRange (0.10 .. 1.00)
bpskInput	CHOICE		EnumRange (IFR AUX-1 AUX-2 PCM-1 PCM-2 PCM-3 PCM-4)
tm.storageId	INTEGER		IntegerRange (0 .. 0)
tm.storageFiles	INTEGER		IntegerRange (0 .. 0)
tmuType	CHOICE		EnumRange (BPSK BPSK-SQUARE PCM/PM PCM/FM)
tp.invert	CHOICE		EnumRange (OFF INV SWAP-Q-INV SWAP-I-INV)
tp.interleave	CHOICE		EnumRange (OFF ON)
tp.encoding	CHOICE		EnumRange (BP-L NRZ-L NRZ-M NRZ-S BP-M BP-S)
filterFm	CHOICE		EnumRange (10MHz 1MHz 100kHz 5MHz 3MHz 1.5MHz 500kHz 300kHz 150kHz 50kHz 30kHz 15kHz 10kHz 5kHz 13MHz NONE)
codeRate	CHOICE		EnumRange (OFF 1/2 1/3 1/4 1/6)
timeOffset	INTEGER		IntegerRange (0 .. 0)
doppler.enable	CHOICE		EnumRange (OFF ON)

name	*type*	*flags*	*range*
demodulationIndex	FLOAT		DoubleRange (0.00 .. 3.14)
transfer.sync	CHOICE		EnumRange (OFF ON)
transfer.frameSize	INTEGER		IntegerRange (10 .. 32000)
timeTag	CHOICE		EnumRange (LAST-BIT-TELEMETRY FIRST-BIT-TELEMETRY FIRST-BIT-SYNC-WORD)
state.ebNo	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
flags.bitSync	BOOLEAN	R/O	BooleanRange R/O (true false)
flags.pskDemod	BOOLEAN	R/O	BooleanRange R/O (true false)
state.frameSync	TEXT	R/O	StringRange R/O
state.ldpcSlice	TEXT	R/O	StringRange R/O
state.bitSlip	TEXT	R/O	StringRange R/O
state.viterbi	TEXT	R/O	StringRange R/O
state.rsCorr	TEXT	R/O	StringRange R/O
state.tmFileNo	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.tmFileSize	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.berViterbi	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.tmFile	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.tmOffset	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.storage	TEXT	R/O	StringRange R/O
state.berRs	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
state.framesGood	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.framesBad	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.freqOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.symbolOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
state.tp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
state.berViterbiQ	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)

name	*type*	*flags*	*range*
state.doppler	TEXT	R/O	StringRange R/O
state.tmFrames	INTEGER	R/O	IntegerRange R/O (0 .. 0)
bd.polynominal	HEX		HexRange (0 .. 0)
bd.register	HEX		HexRange (0 .. 0)
bd.startOffset	INTEGER		IntegerRange (0 .. 255)
ad.polynominal	HEX		HexRange (0 .. 0)
ad.register	HEX		HexRange (0 .. 0)
ad.startOffset	INTEGER		IntegerRange (0 .. 255)
storage.start	TEXT	NOPRESET	StringRange
storage.stop	TEXT	NOPRESET	StringRange
storage.resume	TEXT	NOPRESET	StringRange
resetCounter	TEXT	NOPRESET	StringRange
saveFilter	TEXT	NOPRESET	StringRange
filterNumber	INTEGER	NOPRESET	IntegerRange (0 .. 255)
faults.01	ALARM	R/O	AlarmFlagRange R/O (Bit Synchronizer)
faults.02	ALARM	R/O	AlarmFlagRange R/O (PSK Demodulator)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Viterbi Decoder)
faults.04	ALARM	R/O	AlarmFlagRange R/O (RS Capacity)
faults.05	ALARM	R/O	AlarmFlagRange R/O (RS Interleave)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Frame Check)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Storage File)
faults.08	ALARM	R/O	AlarmFlagRange R/O (Create File/Dir)

9.798 Zodiac-Cortex-CRT

This page describes the *Zodiac-Cortex-CRT* device driver and its device window. The *Zodiac-Cortex-CRT* device driver uses the following tables: the Cortex CRT, Global Cortex CRT and Noise Table.
It allows controlling units general settings, the if inputs and noise generator and shows main status as well as the TMU mapping and available modules.

Individual modules as IFR, IFM, TMU, TMS, TCU,... use a separate driver which has to be added to the same thread in the device setup. It is based on the STI 100013_CRT remote protocol document v 3.31, dated Jan. 10., 2020

Beside this base unit multiple instances of the various functional elements may be configured in the sat-nms M&C to handle all features of the unit. All devices referring to the same physical device must be configured at the same device thread / interface to ensure that TCP requests and commands to the device do not overlap.

The different available module drivers for the CRT are:

- Zodiac-Cortex-CRT-IFM monitors and controls one IF Modulator incl. an embedded frequency converter
- Zodiac-Cortex-CRT-IFR monitors and controls one IF Receiver incl. an embedded frequency converter
- Zodiac-Cortex-CRT-TCU monitors and controls one Telecommand unit
- Zodiac-Cortex-CRT-TMS monitors and controls one Telemetry Simulator
- Zodiac-Cortex-CRT-TMU monitors and controls one Telemetry Unit
- Zodiac-Cortex-CRT-PRJ monitors and controls one Project Specific table

The CRT requires two separate IP connection for monitoring and commanding the device. The device driver establishes the connection to the monitoring port (usually 3000) by terminal server like TTY connection. This connection is permanently active. Address and port are set in the vlc.setup configuration file as for a terminal server serial port.
How to use TLS connection is described below.

Steps to configure the Zodiac-CRT drivers / devices for TLS communication

1. Create a PKCS12 keystore from the cert files read from the BBU:

```
openssl pkcs12 -export -out bbu.p12 \
  -inkey CRT-Services-Client-v2.key -in CRT-Services-Client-v2.cert.pem
```

openssl prompts to select a password for keystore. e.g. crt

2. Copy the keystore file to /home/satnms/certs/ at the M&C

3. Edit the device configuration (client) to tell the M&C about the protocol and port.

```
PORT "TLS:192.168.2.202:53000" PROTOCOL CortexNT
   IDLE 1000
   DEVICE BBU-1 IS Zodiac-Cortex-CRT
   DEVICE BBU-1-IFR-1 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-2 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-3 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFR-4 IS Zodiac-Cortex-CRT-IFR
   DEVICE BBU-1-IFM-1 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-IFM-2 IS Zodiac-Cortex-CRT-IFM
   DEVICE BBU-1-TMU-A IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-B IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-C IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-D IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-E IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-F IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-G IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TMU-H IS Zodiac-Cortex-CRT-TMU
   DEVICE BBU-1-TCU-1 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TCU-2 IS Zodiac-Cortex-CRT-TCU
   DEVICE BBU-1-TMS-1 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-TMS-2 IS Zodiac-Cortex-CRT-TMS
   DEVICE BBU-1-PRJ IS Zodiac-Cortex-CRT-PRJ
```

Important is the TLS in front of the IP address and the port number to use with TLS communication.

4. Restart the M&C and for each BBU-* device edit these setup parameters.

```
config.cmdPort = 53001
address = certs/bbu.p12 crt
```

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Global unit settings and status.
- --- Overview of installed modules
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters

which must be set to make the software talk to the device.

- *parameter --- description*
- address --- empty or certificate and password when using TLS
- cmdPort --- standard port are 3001 or 53001 when using TLS preset
- deviceComment --- An optional text entered here appears as an addition to the tooltip which is shown while the mouse is placed over a device icon in the GUI. The "\n" character sequence is translated into a newline character, splitting the text into multiple lines.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.softwareId	TEXT	R/O	StringRange R/O
info.model	TEXT	R/O	StringRange R/O
info.serial	TEXT	R/O	StringRange R/O
info.temp.masterBoard	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temp.masterFPGA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temp.slaveBoard	INTEGER	R/O	IntegerRange R/O (0 .. 0)
info.temp.slaveFPGA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.deviceComment	TEXT	SAVE SETUP	StringRange
config.readBackDelay	INTEGER	SAVE SETUP	IntegerRange (100 .. 10000)

name	*type*	*flags*	*range*
internal.cmd	TEXT	R/O	StringRange R/O
internal.type	TEXT	R/O	StringRange R/O
internal.ver	TEXT	R/O	StringRange R/O
internal.rev	TEXT	R/O	StringRange R/O
internal.softwareId1	TEXT	R/O	StringRange R/O
internal.softwareId2	TEXT	R/O	StringRange R/O
internal.softwareId3	TEXT	R/O	StringRange R/O
internal.model	HEX	R/O	HexRange R/O (0 .. 0)
internal.shelfId	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.day.c1	HEX	R/O	HexRange R/O (0 .. 0)
internal.day.c4h	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.day.c4	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hour	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.hour.c1	HEX	R/O	HexRange R/O (0 .. 0)
internal.secs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.msecs.c1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.msecs.c2	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
internal.msecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.usecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.nsecs	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.min	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.sec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.tag	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
days	INTEGER	R/O	IntegerRange R/O (0 .. 0)
hours	INTEGER	R/O	IntegerRange R/O (0 .. 0)
minutes	INTEGER	R/O	IntegerRange R/O (0 .. 0)
seconds	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
cmdstat	TEXT	R/O	StringRange R/O
upload.doppler.file	TEXT	NOCOMPARE	StringRange
module.hasTcu1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTcu2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTms1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTms2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTms3	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfm1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfm2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfm3	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfmX1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfmX2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfmX3	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasRau	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasRauE	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfr1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfr2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfr3	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfr4	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfr5	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfrX1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfrX2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfrX3	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfrX4	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasIfrX5	BOOLEAN	R/O	BooleanRange R/O (true false)

name	*type*	*flags*	*range*
module.hasDcu1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasDcu2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuA	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuB	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuC	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuD	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuE	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuF	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuG	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmuH	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxA	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxB	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxC	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxD	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxE	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxF	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxG	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasTmxH	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasNoise	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasEfc	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasAit1	BOOLEAN	R/O	BooleanRange R/O (true false)
module.hasAit2	BOOLEAN	R/O	BooleanRange R/O (true false)
module.ifrTmuA	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuB	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuC	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)

name	*type*	*flags*	*range*
module.ifrTmuD	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuE	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuF	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuG	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
module.ifrTmuH	CHOICE	R/O	EnumRange R/O (IFR-1 IFR-2 IFR-3 IFR-4)
dataflowMon	HEX		HexRange (0 .. 0)
dataflowLog	HEX		HexRange (0 .. 0)
reference	CHOICE		EnumRange (INT 1+INT 2+INT 1+2+INT)
leap	INTEGER		IntegerRange (0 .. 0)
leap.mode	CHOICE		EnumRange (AUTOMATIC MANUAL)
leap.year	CHOICE		EnumRange (YES NO)
leap.second	CHOICE		EnumRange (OFF POSITIVE NEGATIVE)
time.offset	INTEGER		IntegerRange (0 .. 0)
time.nextOffsetDate.1	INTEGER		IntegerRange (0 .. 0)
time.nextOffset.1	INTEGER		IntegerRange (0 .. 0)
time.nextOffsetDate.2	INTEGER		IntegerRange (0 .. 0)
time.nextOffset.2	INTEGER		IntegerRange (0 .. 0)
time.code	CHOICE		EnumRange (IRIG-B NASA-36 NTP)
video.testpoint.1	CHOICE		EnumRange (OFF IFM-1 IFM-2 TCU-1 TCU-2 RAU TMS-1 TMS-2 IFR-1 IFR-2 IFR-3 IFR-4 IFR-5 DCU-1 DCU-2 INTERNAL-1 INTERNAL-2)

name	*type*	*flags*	*range*
video.testpoint.2	CHOICE		EnumRange (OFF IFM-1 IFM-2 TCU-1 TCU-2 RAU TMS-1 TMS-2 IFR-1 IFR-2 IFR-3 IFR-4 IFR-5 DCU-1 DCU-2 INTERNAL-1 INTERNAL-2)
input.1	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.2	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.3	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.4	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.5	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.6	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.1.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
input.2.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
input.3.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
input.4.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
input.5.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
input.6.agc	CHOICE		EnumRange (0.1ms 1ms 10ms 100ms 1000ms)
time.tag	CHOICE	R/O	EnumRange R/O (CODE-0 CODE-1 CODE-2 CODE-3 CODE-4 CODE-5)
crtGlobalTimeTag	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
crtGlobalTimeTag2	TEXT	R/O	StringRange R/O
ctx.document	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
ctx.loadSave	BOOLEAN	R/O	BooleanRange R/O (true false)
state.refFreq	TEXT	R/O	StringRange R/O
state.refInput	TEXT	R/O	StringRange R/O
state.decoder	TEXT	R/O	StringRange R/O
flags.extPps	BOOLEAN	R/O	BooleanRange R/O (true false)
clients.mon	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.ctl	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tms1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tms2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tms3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tcu1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tcu2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.rng	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.meas	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.log	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.spec	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.dopp	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.vect	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.other	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuA	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuB	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuC	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuD	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuE	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuF	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuG	INTEGER	R/O	IntegerRange R/O (0 .. 0)
clients.tmuH	INTEGER	R/O	IntegerRange R/O (0 .. 0)

name	*type*	*flags*	*range*
neuronFlows	INTEGER	R/O	IntegerRange R/O (0 .. 0)
noise.1.on	CHOICE		EnumRange (OFF ON)
noise.2.on	CHOICE		EnumRange (OFF ON)
noise.3.on	CHOICE		EnumRange (OFF ON)
noise.1	FLOAT		DoubleRange (15.0 .. 100.0)
noise.2	FLOAT		DoubleRange (15.0 .. 100.0)
noise.3	FLOAT		DoubleRange (15.0 .. 100.0)
reloadDefault	TEXT	NOPRESET	StringRange
setWinTimeToIrig	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (Remote Access)
faults.02	ALARM	R/O	AlarmFlagRange R/O (External PPS)
faults.03	ALARM	R/O	AlarmFlagRange R/O (Time)
faults.04	ALARM	R/O	AlarmFlagRange R/O (Emulation Mode)
faults.05	ALARM	R/O	AlarmFlagRange R/O (Flash Memory)
faults.06	ALARM	R/O	AlarmFlagRange R/O (Licenses)
faults.07	ALARM	R/O	AlarmFlagRange R/O (Menu Configuration)
faults.08	ALARM	R/O	AlarmFlagRange R/O (FPGA Configuration)
faults.09	ALARM	R/O	AlarmFlagRange R/O (MSP Init)
faults.10	ALARM	R/O	AlarmFlagRange R/O (FPGA Temperature)
faults.11	ALARM	R/O	AlarmFlagRange R/O (Temperature)

name	*type*	*flags*	*range*
faults.12	ALARM	R/O	AlarmFlagRange R/O (Plugin extension)
faults.13	ALARM	R/O	AlarmFlagRange R/O (Driver init)
faults.14	ALARM	R/O	AlarmFlagRange R/O (Memory init)
faults.15	ALARM	R/O	AlarmFlagRange R/O (DL FPGA Init)
faults.16	ALARM	R/O	AlarmFlagRange R/O (UL FPGA Init)
faults.17	ALARM	R/O	AlarmFlagRange R/O (PLL)
faults.18	ALARM	R/O	AlarmFlagRange R/O (IRIG-B/NASA Unlock)
faults.20	ALARM	R/O	AlarmFlagRange R/O (TCU-1)
faults.21	ALARM	R/O	AlarmFlagRange R/O (TCU-2)
faults.22	ALARM	R/O	AlarmFlagRange R/O (TMS-1)
faults.23	ALARM	R/O	AlarmFlagRange R/O (TMS-2)
faults.24	ALARM	R/O	AlarmFlagRange R/O (TMS-3)
faults.25	ALARM	R/O	AlarmFlagRange R/O (IFM-1)
faults.26	ALARM	R/O	AlarmFlagRange R/O (IFM-2)
faults.27	ALARM	R/O	AlarmFlagRange R/O (IFM-3)
faults.28	ALARM	R/O	AlarmFlagRange R/O (IFR-1)
faults.29	ALARM	R/O	AlarmFlagRange R/O (IFR-2)
faults.30	ALARM	R/O	AlarmFlagRange R/O (IFR-3)
faults.31	ALARM	R/O	AlarmFlagRange R/O (IFR-4)
faults.32	ALARM	R/O	AlarmFlagRange R/O (IFR-5)
faults.33	ALARM	R/O	AlarmFlagRange R/O (TMU-A)
faults.34	ALARM	R/O	AlarmFlagRange R/O (TMU-B)
faults.35	ALARM	R/O	AlarmFlagRange R/O (TMU-C)

name	*type*	*flags*	*range*
faults.36	ALARM	R/O	AlarmFlagRange R/O (TMU-D)
faults.37	ALARM	R/O	AlarmFlagRange R/O (TMU-E)
faults.38	ALARM	R/O	AlarmFlagRange R/O (TMU-F)
faults.39	ALARM	R/O	AlarmFlagRange R/O (TMU-G)
faults.40	ALARM	R/O	AlarmFlagRange R/O (TMU-H)
faults.41	ALARM	R/O	AlarmFlagRange R/O (TMU-A Program)
faults.42	ALARM	R/O	AlarmFlagRange R/O (TMU-B Program)
faults.43	ALARM	R/O	AlarmFlagRange R/O (TMU-C Program)
faults.44	ALARM	R/O	AlarmFlagRange R/O (TMU-D Program)
faults.45	ALARM	R/O	AlarmFlagRange R/O (TMU-E Program)
faults.46	ALARM	R/O	AlarmFlagRange R/O (TMU-F Program)
faults.47	ALARM	R/O	AlarmFlagRange R/O (TMU-G Program)
faults.48	ALARM	R/O	AlarmFlagRange R/O (TMU-H Program)
faults.49	ALARM	R/O	AlarmFlagRange R/O (Noise Generator)
faults.50	ALARM	R/O	AlarmFlagRange R/O (Noise Channel 1)
faults.51	ALARM	R/O	AlarmFlagRange R/O (Noise Channel 2)
faults.52	ALARM	R/O	AlarmFlagRange R/O (Noise Channel 3)

9.799 Zodiac-Cortex-DTR-Cal

This page describes the device driver and the device window for the*Zodiac-Cortex-DTR-Cal* device. This device supports the phase calibration of a Zodiac DTR together with a HITEC monopulse ACU. The calibration values determined in the ACU are copied into the memory of the DTR.

This very special device works only together with the Zodiac-Cortex-DTR and the Hitec-HACU-01 devices.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains only a button to triggere the copy process
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
acuName	The device name of the ACU.
dtrName	The device name of the DTR

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
config.acuName	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.dtrName	TEXT	SAVE SETUP	StringRange
setCalibration	TEXT	NOPRESET	StringRange

9.800 Zodiac-Cortex-DTR-Path

This page describes the device driver and the device window for the*Zodiac-Cortex-DTR-Path* device. This device sets the signal path control word at a Zodiac Cortex DTR.

The device encodes an eight character configuration string into a 32 bit binary word and commands this to the DTR. It uses a special implementation of the communication protocol which only connects to the DTR to send the command and after this disconnects again. This is because the DTR accepts only one connection at a time.

The eight characters of the configuration string may either be set at the main screen of the device or the device may be configured to subscribe these values from some other devices like Gate-Arrays .

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Sets / shows the characters of the configuration string
- --- The faults page.
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	*description*
address	The IP address of the Zodiac Cortex device to connect to.
portNumber	The IP port number to connect to. For the primary DTR this is 3080 (the default value), for the secondary DTR set 3081.

parameter	*description*
charId.0 charId.1 charId.2 charId.3 charId.4 charId.5 charId.6 charId.7	The parameter IDs of the variables where to read the up to 8 characters for the DTR configuration string. These typically are the outputs of Gate-Array devices which encode some waveguide switch poistions to the characters of the configuration string. *charId.0* denotes the leftmost character of the string, *charId.7* the rightmost one.

Remarks

The first two characters of the configuration string have fixed values, they are reserved for future use. The driver initializes them to ".".

On startup, the *Zodiac-Cortex-DTR-Path* device will not send a configuration command to the DTR until the configuration string is complete.

If the configuration string uses the same status character at two positions, simply configure the *charId.** settings for both positions to the same Gate-Array output.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.startup	TEXT	R/O	StringRange R/O
config.portNumber	INTEGER	SAVE SETUP	IntegerRange (0 .. 65535)
config.charId.0	TEXT	SAVE SETUP	StringRange
config.charId.1	TEXT	SAVE SETUP	StringRange
config.charId.2	TEXT	SAVE SETUP	StringRange
config.charId.3	TEXT	SAVE SETUP	StringRange
config.charId.4	TEXT	SAVE SETUP	StringRange
config.charId.5	TEXT	SAVE SETUP	StringRange
config.charId.6	TEXT	SAVE SETUP	StringRange

name	*type*	*flags*	*range*
config.charId.7	TEXT	SAVE SETUP	StringRange
address	TEXT	SAVE	StringRange
char.0	TEXT		StringRange
char.1	TEXT		StringRange
char.2	TEXT		StringRange
char.3	TEXT		StringRange
char.4	TEXT		StringRange
char.5	TEXT		StringRange
char.6	TEXT		StringRange
char.7	TEXT		StringRange
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O

9.801 Zodiac-Cortex-DTR

This page describes the device driver and the device window for the*Zodiac-Cortex-DTR* device. This device monitors and controls a limited number of parameters of the inSnec/Zodiac digital tracking receiver

The fact, that the device uses two separate TCP connections for monitoring and controlling the device, care must be taken to setup the IP address and the port numbers correctly.

The connection for monitoring is setup like a terminal server serial port: In the device setup window, the device's IP address and the monitoring port number (typically 3000) are entered as the interface name, separated by a colon character. Example "192.168.0.1:3000".

The port number for commanding the device (typically 3001) is set as a configuration parameter of the device driver as described below.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- Contains the DTR parameters and readings.
- --- The faults page.

- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to integrate the device into the VLC/M&C.

parameter	***description***
address	May be left empty.
cmdPort	The IP port number used to command the device. This typically is 3001.

Variables defined by this device driver

name	***type***	***flags***	***range***
info.driver	TEXT	R/O	StringRange R/O
lowLevel.cmd	TEXT		StringRange
lowLevel.reply	TEXT	R/O	StringRange R/O
faults.99	ALARM	R/O	AlarmFlagRange R/O (Communication)
faults.commstat	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O
info.frame	TEXT	R/O	StringRange R/O
info.revision	TEXT	R/O	StringRange R/O
info.tracking	TEXT	R/O	StringRange R/O
config.cmdPort	INTEGER	SAVE SETUP	IntegerRange (0 .. 0)
config.noIfPorts	CHOICE	SAVE SETUP	EnumRange (1 2 3 4 5 6)
config.noDcu	CHOICE	SAVE SETUP	EnumRange (0 1 2)
internal.a	TEXT	R/O	StringRange R/O
internal.b	TEXT	R/O	StringRange R/O

name	*type*	*flags*	*range*
internal.c	TEXT	R/O	StringRange R/O
internal.d	TEXT	R/O	StringRange R/O
internal.manElPhA	TEXT	R/O	StringRange R/O
internal.autElPhA	TEXT	R/O	StringRange R/O
internal.manElGaA	TEXT	R/O	StringRange R/O
internal.autElGaA	TEXT	R/O	StringRange R/O
internal.manAzGaA	TEXT	R/O	StringRange R/O
internal.autAzGaA	TEXT	R/O	StringRange R/O
internal.manElPhB	TEXT	R/O	StringRange R/O
internal.autElPhB	TEXT	R/O	StringRange R/O
internal.manElGaB	TEXT	R/O	StringRange R/O
internal.autElGaB	TEXT	R/O	StringRange R/O
internal.manAzGaB	TEXT	R/O	StringRange R/O
internal.autAzGaB	TEXT	R/O	StringRange R/O
internal.aliasPol	TEXT	R/O	StringRange R/O
internal.pos1	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos2	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.pos3	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.94	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.95	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.96	INTEGER	R/O	IntegerRange R/O (0 .. 0)
internal.cmd	TEXT	R/O	StringRange R/O
frequency	FLOAT		DoubleRange (0.000000 .. 0.000000)
input	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.1	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.2	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.3	CHOICE		EnumRange (NOMINAL ALTERNATE)

name	*type*	*flags*	*range*
input.4	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.5	CHOICE		EnumRange (NOMINAL ALTERNATE)
input.6	CHOICE		EnumRange (NOMINAL ALTERNATE)
dcu.1	CHOICE		EnumRange (A B A+B)
dcu.2	CHOICE		EnumRange (A B A+B)
operatingMode	CHOICE		EnumRange (AUTO MANUAL)
carrierAcquisition	CHOICE		EnumRange (NORMAL HIGH)
bandwidth	CHOICE		EnumRange (30Hz 100Hz 300Hz 1000Hz 3000Hz)
agcTimeConstant	CHOICE		EnumRange (1ms 10ms 100ms)
carrierOffset	FLOAT	R/O	DoubleRange R/O (0.000 .. 0.000)
level	FLOAT	R/O	DoubleRange R/O (0.00 .. 0.00)
monopulseAzError	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
monopulseElError	FLOAT	R/O	DoubleRange R/O (0.00000 .. 0.00000)
phasingMode	CHOICE		EnumRange (AUTO MANUAL)
inputFrequency	FLOAT		DoubleRange (500.000 .. 40000.000)
systemConfigurationA	HEX	R/O	HexRange R/O (0 .. 0)
elevationPhaseA	INTEGER		IntegerRange (0 .. 0)
elevationGainSlopeA	FLOAT		DoubleRange (-250.00000 .. 250.00000)
azimuthGainSlopeA	FLOAT		DoubleRange (-250.00000 .. 250.00000)
reverseAzError	CHOICE		EnumRange (NO YES)
systemConfigurationB	HEX	R/O	HexRange R/O (0 .. 0)
elevationPhaseB	INTEGER		IntegerRange (0 .. 0)
elevationGainSlopeB	FLOAT		DoubleRange (-250.00000 .. 250.00000)

name	*type*	*flags*	*range*
azimuthGainSlopeB	FLOAT		DoubleRange (-250.00000 .. 250.00000)
reverseAzErrorB	CHOICE		EnumRange (NO YES)
storePhaseTable	TEXT	NOPRESET	StringRange
faults.01	ALARM	R/O	AlarmFlagRange R/O (IF Lock)

9.802 Zodiac-Cortex-Reset

This page describes the device driver and the device window for the*Zodiac-Cortex-Reset* device.

Device window pages

The following table shows which device window pages are available with this individual device type. Tool-bar functions not mentioned here are described at the general description of device windows .

- --- The operational parameters and meter readings
- --- The device info page.
- --- The maintenance page. The configuration parameters described below are set at this page.

Configuration parameters

At the maintenance page of the device window there are a couple of configuration parameters which must be set to make the software talk to the Amplifier.

parameter	*description*
address	The IP address of the Zodiac Cortex Reset to connect.
portNumber	The IP port number used to command the device. This typically is 3002.

Variables defined by this device driver

name	*type*	*flags*	*range*
info.driver	TEXT	R/O	StringRange R/O
info.type	TEXT	R/O	StringRange R/O
info.port	TEXT	R/O	StringRange R/O

name	***type***	***flags***	***range***
info.frame	TEXT	R/O	StringRange R/O
config.portNumber	INTEGER	SAVE SETUP	IntegerRange (0 .. 65535)
reset	TEXT	NOPRESET	StringRange